

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství,



Diplomová práce

**Návrh modelu dynamické cenotvorby
v prostředí online prodeje**

Bc. Tomáš Braha

© 2021 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Tomáš Braha

Kvantitativní metody v ekonomice

Systémové inženýrství

Název práce

Návrh modelu dynamické cenotvorby v prostředí online prodeje

Název anglicky

Dynamic pricing in e-commerce

Cíle práce

Cílem diplomové práce je vytvoření postupu dynamické cenotvorby pro online prodej nehmotného produktu za účelem maximalizace z něj plynoucích tržeb.

Metodika

Teoretická část bude zaměřena na popis specifik pricingu v prostředí online prodeje nehmotných produktů a dále na popis umělé inteligence, zejména pak výhod její aplikace v modelech neuronových sítí, které lze využít právě k dynamické cenotvorbě. Čerpáno bude z odborných zdrojů – monografií, vědeckých publikací a případně též z relevantních webových stránek. Na základě rešerše budou vytvořeny vhodné metody pro rozbor a úpravu souboru dat do modelu neuronových sítí. Vzhledem k jejich rozsáhlosti budou využity nástroje jako např. Google Analytics 360, MS Excel či PowerBI. Samotné výpočty budou provedeny v prostředí softwaru Azure Machine Learning Studio.

V praktické části budou na základě navržené metody provedeny potřebné modelové experimenty. Výsledky modelu budou ověřeny za pomoci simulace scénářů prodeje s různými cenami produktu. Tím bude zhodnocena jeho efektivita a využitelnost s ohledem na definovaný cíl maximalizace tržeb.

Doporučený rozsah práce

70

Klíčová slova

umělá inteligence, dynamická cenotvorba, e-commerce, modely neuronových sítí, maximalizace tržeb, pricing

Doporučené zdroje informací

CASTROUNIS, Alex, 2019. AI for People and Business. USA: O'Reilly Media. ISBN 9781492036579.
POKORNÝ, Miroslav, 1996. Umělá inteligence v modelování a řízení. ČR, Praha: BEN – technická literatura. ISBN 80-901984-4-9.
ROY, Samir a Udit CHAKRABORTY, 2013. Soft Computing. India: Pearson Education. ISBN 9789332514201.
SIDDIQUE, Nazmul a Hojjat ADELI, 2013. Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks a Evolutionary Computing. United States: John Wiley. ISBN 9781118337844.
VESELÝ, Arnošt, 2012. Metody umělé inteligence. ČR, Praha: Česká zemědělská univerzita, Provozně ekonomická fakulta. ISBN 978-80-213-2295-0.

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

prof. RNDr. Helena Brožová, CSc.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 19. 1. 2021

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 22. 1. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 31. 03. 2021

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh modelu dynamické cenotvorby v prostředí online prodeje" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31. 3. 2021

Poděkování

Rád bych touto cestou poděkoval především prof. RNDr. Heleně Brožové, CSc., za odborné vedení práce a také mému nejbližšímu okolí za podporu a trpělivost.

Návrh modelu dynamické cenotvorby v prostředí online prodeje

Abstrakt

Cílem této diplomové práce je vytvořit postup dynamické cenotvorby pro online prodej nehmotného produktu za účelem maximalizace tržeb. K tomu využívá umělé inteligence, konkrétně rekurentních neuronových sítí, které si dokážou zapamatovat data z předešlých kroků výpočtů a hodí se tak pro analýzu časových řad. V rámci tohoto je pro samotnou práci použit model Long-short term memory, ke kterému je přistoupeno přes programovací knihovny využívajících jazyka Python. Tento model dokáže stanovit predikční hodnotu na základě sekvenčních dat. Jako vstup do toho modelu jsou použity hodnoty o průměrných příjmech za transakci, na jejichž model vytváří predikce. Ty autor navrhuje použít pro rozhodování o stanovení ceny a toto blíže popisuje s ohledem na specifický přístup oddělení firmy, ze které tato data pochází. Autor si je vědom zjednodušení a opomenutí některých dalších parametrů, avšak svou prací potvrzuje, že jde o směr, kterým je vhodné se při řešení této problematiky zabývat.

Klíčová slova: dynamická cenotvorba, umělá inteligence, rekurentní neuronové sítě, model Long-short term memory, e-commerce, average order value, Python

Dynamic pricing in e-commerce

Abstract

The aim of this thesis is to create a dynamic pricing process for online sales of intangible products in order to maximize sales. To do this, it uses artificial intelligence, specifically recurrent neural networks, which can memorize data from previous calculation steps and are therefore suitable for time series analysis. Within this, the Long-short term memory model is used for the work itself and which is accessed via programming libraries based on the Python language. This model can determine the prediction value on the basis of sequence data. As an input to this model, values of average revenues per transaction are used, of which model makes predictions. The author suggests using them to make decisions on pricing and describes this in more detail with regard to the specific approach of the department of the company from which this data comes. The author is aware of the simplification and omission of some other parameters, but his work confirms that this is the direction in which it is appropriate to address this issue.

Keywords: dynamic pricing, artificial intelligence, recurrent neural networks, Long-short term memory model, e-commerce, average order value, Python

Obsah

1	Úvod.....	1
2	Cíl práce a metodika	2
2.1	Cíl práce	2
2.2	Metodika	3
2.2.1	Použité nástroje.....	4
2.2.2	Programovací jazyk Python a použité knihovny	5
2.2.3	Postup řešení úloh neuronových sítí	7
3	Teoretická východiska	8
3.1	Umělá inteligence.....	8
3.1.1	Historie umělé inteligence	8
3.1.2	Turingův test a AI	9
3.1.3	Česko a AI	9
3.1.4	Soft Computing.....	10
3.1.5	Neuron	11
3.1.6	Učení.....	15
3.1.7	Neuronové sítě	16
3.1.8	Rekurentní neuronové sítě	19
3.1.9	Model Long-short term memory.....	19
3.1.10	Posouzení kvality modelu.....	23
3.1.11	Přeučení sítě.....	24
3.2	Základní metriky používané v e-commerce.....	25
3.2.1	Tržby.....	25
3.2.2	Transakce	26
3.2.3	Average Order Value	26
3.3	Časové řady.....	26
3.3.1	Základní charakteristiky	27

3.3.2	Regresní modely	27
3.3.3	Regrese a neuronové sítě	28
3.3.4	Chyba předpovědi	28
4	Vlastní práce	30
4.1	Základní analýza dat	30
4.1.1	Získání dat.....	30
4.1.2	Základní analýza dat	30
4.2	Výpočet pomocí Microsoft Azure Machine Learning Studia	34
4.2.1	Trénovací experiment	34
4.2.2	Prediktivní experiment.....	35
4.3	Výpočet modelu LSTM.....	36
4.3.1	Příprava dat	36
4.3.2	Příprava časové řady a výpočet AOV	39
4.3.3	Trénování modelu	42
4.3.4	Výpočet modelu LSTM	44
4.4	Simulace	49
4.4.1	Změna počtu neuronů ve skrytých vrstvách	50
4.4.2	Pozměněná konfigurace sítě	59
5	Výsledky a diskuze	62
5.1	Výsledky základní analýzy dat.....	62
5.1.1	Transakce a tržby	62
5.1.2	AOV.....	62
5.1.3	Faktory ovlivňující cenu kupónu	62
5.2	Výsledky modelu LSTM.....	63
5.2.1	Výsledky simulace modelu LSTM	63
5.2.2	Sestavení modelu pro tvorbu ceny v prostředí online prodeje	65
6	Závěr	67

7	Seznam použitých zdrojů.....	68
8	Přílohy.....	73

Seznam obrázků

Obrázek 1:	Turingův test	9
Obrázek 2:	Popis biologického neuronu	11
Obrázek 3:	Formální model neuronu	12
Obrázek 4:	Aktivační funkce typu sigmoida (vlevo) a hyperbolický tangent (vpravo)	14
Obrázek 5:	Aktivační funkce typu ReLU	14
Obrázek 6:	Dopředná síť s jednou vrstvou	17
Obrázek 7:	Dopředná síť s více vrstvami	18
Obrázek 8:	Síť s rozšířenou skrytou vrstvou (vlevo) a síť pyramidální (vpravo)	18
Obrázek 9:	Ukázka rekurentní neuronové sítě.....	19
Obrázek 10:	Schéma modelu LSTM	20
Obrázek 11:	Forget gate layer.....	21
Obrázek 12:	Proces ukládání informací do cell state.....	21
Obrázek 13:	Zápis hodnot do state	22
Obrázek 14:	Výstup z buňky	22
Obrázek 15:	Přeučení sítě	25
Obrázek 17:	Diagram trénovacího experimentu	35
Obrázek 18:	Diagram prediktivního experimentu	35
Obrázek 19:	Schéma nastavení cen pro finálního zákazníka.....	66

Seznam grafů

Graf 1:	Špatná predikční schopnost	23
Graf 2:	Uspokojivá predikční schopnost	23
Graf 3:	Velmi dobrá predikční schopnost.....	24
Graf 4:	Tržby v letech 2016–2020.....	31
Graf 5:	Sumarizace tržeb v jednotlivých letech.....	31
Graf 6:	Transakce (ks) v letech 2016 - 2020	32
Graf 7:	Sumarizace transakcí (ks) za jednotlivé roky	32
Graf 8:	AOV v letech 2016 - 2020	33

Graf 9: Přehled boxplot grafů pro jednotlivé roky	34
Graf 10: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 4-8-16-1	50
Graf 11: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 4-8-16-1.....	51
Graf 12: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 4-16-8-1	51
Graf 13: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 4-16-8-1.....	52
Graf 14: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 8-16-32-1	52
Graf 15: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 8-16-32-1.....	53
Graf 16 : Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 8-32-16-1.....	54
Graf 17 : Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 8-32-16-1.....	54
Graf 18: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 16-32-64-1	55
Graf 19: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 16-32-64-1.....	55
Graf 20: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 16-64-32-1.....	56
Graf 21: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 16-64-32-1.....	56
Graf 22: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 32-64-128-1.....	57
Graf 23: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 32-64-128-1.....	57
Graf 24: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě 32-128-64-1.....	58
Graf 25: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 32-128-64-1.....	58
Graf 26: Znárodnění ztráty modelu v průběhu epoch s přidáním jednou skrytou vrstvou ..	59
Graf 27: Znárodnění ztráty modelu v průběhu epoch pro konfiguraci sítě s ubránou jednou skrytou vrstvou	60
Graf 28: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě s přidáním jednou skrytou vrstvou	60
Graf 29: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě s ubránou jednou skrytou vrstvou	61

Seznam tabulek

Tabulka 1: Tvrdé a měkké metody výpočtů	10
Tabulka 2: Implementace používaných knihoven	36
Tabulka 3: Nastavení šířky a výšky grafů	37
Tabulka 4: Načtení csv souboru.....	37
Tabulka 5: Zobrazení souborů dat „df“	37
Tabulka 6: Základní informace o souboru „df“	38
Tabulka 7: Výpis základních charakteristik souboru df	39

Tabulka 8: Výpočet AOV	39
Tabulka 9: Změna formátu datumu, převedení na index a zobrazení informací o souboru	40
Tabulka 10: Grafické zobrazení AOV	41
Tabulka 11: Charakteristiky souboru dat df2	41
Tabulka 12: Rozdělení souboru df2 na trénovací a testovací	42
Tabulka 13: Použití funkce MinMaxScaler()	42
Tabulka 14: Sekvenční postup výpočtu	43
Tabulka 15: Algoritmus pro vytvoření sekvencí	43
Tabulka 16: Zobrazení struktury trénovacího a testovacího souboru	44
Tabulka 17: Algoritmus pro výpočet modelu LSTM	45
Tabulka 18: Algoritmus pro zobrazení ztráty modelu	46
Tabulka 19: Načtení modelu a predikce hodnot	47
Tabulka 20: Inverzní transformace hodnot	48
Tabulka 21: Zobrazení grafu predikovaných a skutečných hodnot	49
Tabulka 22: Přehled výsledků simulací	64

1 Úvod

Problematika stanovení cen je stará jako sám obchod realizovaný prostřednictvím platidla. Zjednodušeně řečeno prodávající se snaží o cenu co nejvyšší, kupující naopak co nejnižší. Zatímco východní společnosti stále inklinují ke stanovování ceny smlouváním, západní svět je zvyklý na předem dané ceny. Příchod podnikání a obchodování na internetu, označované již zdomácněným anglickým výrazem e-commerce, tuto otázku proměnlivé ceny opět oživilo. Umožňuje totiž okamžitě a pružně reagovat na konkrétního zákazníka, aniž by do toho jakkoli prodejce musel zasahovat. Rozvoj umělé inteligence toto ještě více podpořil tím, že dal online marketérům do rukou nástroje zvyšující efektivitu jejich snažení. Umělá inteligence je bezesporu disruptivní vynález, který mění léta zaběhlé postupy. Dnes je možné se s ní setkat na mnoha místech, ať už na webu ve formě různých chatbotů, v internetové televizi při přizpůsobování obsahu sledujícímu nebo třeba v chytrých telefonech při nabídce slov při psaní věty. To všechno je možné, neboť umělá inteligence umožňuje snadnou analýzu obrovského množství dat. Její neuronová síť sama dokáže hledat zákonitosti v předložených datech a porovnávat je mezi sebou. Díky tomu může být pak schopná vytvořit prediktivní model, jenž předpovídá budoucí vývoj.

Zmíněné schopnosti umělé inteligence se nabízí využít i v rozhodovacích modelech, kdy je možné na základě původních hodnot vydat doporučení budoucích rozhodnutí. V rámci zmíněné cenotvorby je při hledání nejvhodnější ceny nutné dbát především na dynamičnost trhu, která je podmíněna rychle se měnícím světem a silnou konkurencí.

Cílem této diplomové práce je vytvořit postup dynamické cenotvorby pro online prodej nehmotného produktu za účelem maximalizace tržeb. Autor si zvolil právě tento druh produktu, neboť pracuje pro společnost, která jej vytváří a prodává přes internet, a může tak využít dat o prodeji konkrétního druhu licence pro prověření využitelnosti umělé inteligence pro dynamickou cenotvorbu. Konkrétně jde o prodej softwaru VPN¹, který se používá k ochraně komunikačních zařízení. Firemní oddělení, která mají jeho prodej na starost, byla v roce 2020 pod tlakem, aby využila velkého zájmu o dovybavení počítačů při náhlém přechodu mnoha pracovníků na práci z domova. Přitom však neexistovala jasná strategie prodeje jednotlivých kupónů, zejména co se týče jejich ceny. Tato volba zaměření diplomové práce tak zároveň vytváří potenciál, že její závěry mohou najít i své praktické využití.

¹Pozn. autora: Virtual Private Network

2 Cíl práce a metodika

2.1 Cíl práce

Jak již z úvodní části vyplývá, hlavním cílem je vytvoření takového modelu, který dokáže predikovat hodnotu ceny kupónu, podle které je možné stanovovat prezentovanou cenu.

Při tomto procesu je potřeba si položit tyto základní otázky:

1. Jaká je optimální cena, za kterou by se vybraný kupón dal prodávat v různém časovém horizontu?
2. Jak moc je možné se od dané ceny odchýlovat?

Autor se domnívá, že právě využití predikčních modelů na základě umělých neuronových sítí by mohlo dopomoci různým oddělením v rozhodování o cenách, které by dále distribuovali a prezentovali konečným zákazníkům. Jedná se především o tým podpory a e-commerce. Tato dvě oddělení se nejvíce podílí na změnách finálních prezentovaných cen. V prostředí e-commerce pak jde především o kanály webového rozhraní, mezi něž spadá web-messaging (dále jen WM), hero a stránka s nabízenými kupóny (nazývaná obecně jako pricing page, dále jen PP). Hero je hlavní část webové stránky, která nabízí maximální prezentovanou slevu. Vyskytuje se na hlavních stránkách (především pak index page, dále jen IP). V týmu podpory se potom jedná o komunikaci s frustrovanými zákazníky, kteří na konci komunikačního procesu očekávají kompenzace za způsobený problém. To se netýká všech zákazníků. Tým podpory má ve své podstatě „volnou ruku“ v distribuování slev pro tyto frustrované zákazníky. Dosud není známa podrobná strategie či alespoň souhrn, jak se má s těmito slevami nakládat. Potom ale nastává problém značného zatížení tvorby ceny individuálním přístupem. V některých případech docházelo dokonce i k prodeji ročních licencí za méně než 10 USD. Běžná cena ročního kupónu se nyní pohybuje okolo 50 USD. V neposlední řadě dochází i ke změnám prezentovaných cen na PP, kde se ceny tolik nemění. WM a hero pak často slouží k prezentování akčních cen. Takto obecně lze shrnout hlavní kanály prezentace a distribuce kupónů.

2.2 Metodika

Je celá řada možností, jak se nad výše uvedenou problematikou v online prostředí zamýšlet. Řada společností používá různé modely, jež jsou specificky přizpůsobeny vzhledem k možnostem vstupních dat a zároveň i k právním hlediskům daných zemí.

Například Evropská unie takto rychle se měnící ceny povoluje, ale zároveň jasně zdůrazňuje, že subjekt nesmí být diskriminován. To nastává, pokud se přitom pracuje s údaji například o věku, pohlaví či sexuální orientaci. Podrobnější zprávu lze najít na stránkách Evropské komise (Jourová, 2016).

V návaznosti na tuto zprávu nepřipadá v úvahu pracovat s datovými soubory, které některé z těchto informací obsahují, respektive nelze se na jejich základě rozhodovat. To do určité míry snižuje možnost modely přizpůsobit co nejlépe potenciálnímu zákazníkovi, ale beze sporu je to správná věc z hlediska svobody spotřebitele.

Cena se ale smí měnit například na základě státu (v prostředí e-commerce někdy označované anglickým „locale“). Lze tak prodávat v některých státech levněji, či draž. To většinou závisí na specifikaci trhu jako celku a konkurenci působící na daném trhu. Zabývat se všemi státy ve kterých společnost nabízí svůj produkt, kterých je opravdu celá řada (k prosinci minulého roku jich bylo 52), je nad rámec této práce. A proto byl vybrán jen jeden stát čili USA. Ten je zvolen především vzhledem k vysoké fluktuaci hodnot v datových souborech. Zároveň se jedná o největší trh, kde firma působí.

Jednou z dalších problematik je přístupnost k potřebným souborům obsahujících „citlivá“ data, obzvláště pokud se jedná o obsáhlé soubory dat, které sumarizují prodeje na dominantních trzích jako je USA. Obecně platí, že firmy, které kvalitně pracují s daty mají jistou konkurenční výhodu. To ale také vede ke zvyšování nákladů na správu těchto dat. Nehledě na odborníky, které firma musí být schopna zaplatit, či neustále rostoucí potřebu navyšovat uložení, kde jsou data ukládána. Každý unik dat může firmu velice poškodit, proto v dané firmě není povolena jakákoliv manipulace s daty bez souhlasu, či přístupového práva. Zároveň se data nesmí v celé míře publikovat veřejně. Tento problém se týká právě požadovaných dat. Ta byla nakonec získána, ale byla záměrně pozměna.

Datové soubory, které jsou potřebné pro tuto práci jsou záznamy o tržbách a záznamy o provedených ukončených transakcích, evidované jako časová řada. Z nich je možné vypočítat AOV, které poslouží jako vstup do vybraného modelu umělých neuronových sítí.

2.2.1 Použité nástroje

Pro výpočet je využito moderních nástrojů, v kterých se modely umělých neuronových sítí dále řeší. Mezi ně patří Microsoft Azure Machine Learning Studio (dále jen MAMLS). Dále je pracováno s Jupyter, jež využívá programovací jazyce Python.

Pro analýzu a základní statistické charakteristiky souborů dat je pracováno především s MS Excel, a i vzhledem k obsáhlým výpočtům a potřebnou rychlou vizualizaci dat i se softwarem Microsoft PowerBI.

2.2.1.1 Google Analytics 360

Google Analytics 360 je, spíše než software, především cloudový nástroj pro správu dat. Je velice populární právě ve firmách zabývajících se obchodováním na internetu. Jak již z názvu vyplývá Google Analytics 360 (dále jen GA360) patří firmě Google LLC, jež je jednou z největších technologických firem působících na internetu, a i začlenění jejích nástrojů může mít dopad na tržby. Obzvláště u firem, které díky své strategii přešly plně k prodeji licencí pouze elektronickou cestou. Prostřednictvím přidružených nástrojů, jako jsou Google Ads či Google Optimize, se například testují či propagují webové stránky. Všechny záznamy plynoucí z pohybu zákazníka na webových stránkách nebo i mimo ně, a celou řadu dalších údajů lze pak nalézt právě v GA360. K jednotlivým databázím je nezbytné mít udělené přístupové právo.

2.2.1.2 Microsoft Power BI

V samotném GA360 není povoleno zobrazení některých stěžejních metrik. I samotná vizualizace dat je velice limitovaná a zbytečně náročná, což vedlo k velkému zatížení zaměstnanců, kteří byli odkázáni stahovat data a dále s nimi pracovat v MS Excel, jež pro práci s obsáhlými datovými soubory není stavěný. Proto se v posledních letech začalo využívat speciálních nástrojů pro správu velkého množství dat. Jedním z nich je právě Microsoft Power BI (dále jen MPBI). Výhodou tohoto softwaru, který existuje v podobě desktopové verze, ale zároveň je propojený s cloudovým prostředím pro další úpravy či sdílení, je jeho možnost pracovat s různými zdroji dat. Lze také propojit, mimo jiné, i s vlastními databázemi MAMLS a GA360, což je velice přínosné pro zrychlení práce s daty. MPBI je zároveň propojený s MS Excel anebo MS Powerpoint pro rychlé vytvoření prezentací vybraných vizualizovaných dat. Dokáže také převádět grafy a data prostřednictvím kontingenčních tabulek či grafů do MS Excel. V případě této diplomové

práce je nástroj mimo jiné využit i pro výpočty například maximálních či minimálních hodnot, směrodatné odchylky a hlavně RMSE².

2.2.1.3 Microsoft Azure Machine Learning Studio

Poměrně novým nástrojem od společnosti Microsoft je Microsoft Azure Machine Learning Studio (dále jen MAMLS). Jak vyplývá z názvu, tento nástroj se nachází v prostředí Azure. MAMLS se používá pro jednoduchou tvorbu a správu modelů různých oborů. Datoví vědci mohou uvítat práci s regresními modely, rychlou analýzu vybraných souborů nebo například i modely pro odhalování odchylek v časových řadách. Na druhou stranu jsou zde dostupné modely pracující s rozpoznáváním jazyků a práci s nimi, nebo práci s klasifikací atd. Toho mohou využít například vývojáři aplikací. Dále je umožněno pracovat s modely v testovacím rozhraní a správou API klíčů. Výhodou je zároveň rychlost výpočtů v případě obsáhlých souborů. Velice zajímavé je i samotné prostředí této služby. Je přizpůsobeno i naprostým začátečníkům, ale na druhou stranu umožňuje i programování v jazyce R či Python. Tato služba je však placená. Microsoft nabízí vyzkoušení služby i studentům a díky připsanému kreditu mohou užívat placené funkce této služby.

V této práci je pracováno s jediným možným nabízeným regresním modelem využívající algoritmu neuronových sítí.

2.2.1.4 Jupyter Notebook

Jedním z hlavních použitých nástrojů pro výpočet modelu LSTM (KAPITOLA) je Jupyter Notebook (dále jen JN). Jedná se o webovou aplikaci s otevřeným zdrojovým kódem, která umožňuje vytvářet a sdílet dokumenty, které obsahují živý kód, rovnice, vizualizace a narativní text. Využívá se pro práci s daty, statistické modelování, vizualizaci dat, rovněž i strojové učení a jiné. Novější obdobou je JupyterLab (Jupyter, 2021). JN umožňuje programovat v jazyce Python.

2.2.2 Programovací jazyk Python a použité knihovny

V průběhu programování daného problému budou využity celé řady knihoven, konkrétně TensorFlow, Keras, Pandas, NumPy a Matplotlib.

² Pozn. autora: Root mean square error (někdy česky překládána jako střední kvadratická odchylka)

2.2.2.1 TensorFlow

TensorFlow (dále jen TF) je open-source knihovna vyvinuta týmem Google Brain pro machine learning. Její nástroje pomáhají především datovým vědcům pracovat s daty z hlediska jejich trénování či právě na využití modelu neuronových sítí. Díky této knihovně je dále pracováno s modelem LSTM, jenž je jedním z nejvyužívanějších modelů pro predikci akciových trhů, respektive časových řad. Výhodou knihovny TF je i práce se sekvenčními daty, jež plně podporuje (TensorFlow, 2021).

2.2.2.2 Keras

Keras je knihovna, která poskytuje rozhraní pro umělé neuronové sítě. Je jedna z nejpopulárnějších pro svou jednoduchost a uživatelskou přívětivost. Podporuje back-endová výpočetní jádra neuronových sítí, umožňuje modularitu, snadnou rozšiřitelnost a je kompatibilní s Pythonem, takže umožňuje přidávání nových modulů, které jsou v tomto jazyce napsány (AIWorld, 2019).

2.2.2.3 Pandas

Knihovna Pandas je rovněž jednou z nejvyužívanějších knihoven pro práci s datovými soubory. Její oblíbenost je především v rychlém výčtu důležitých charakteristik souboru. Například obsahuje funkce, které umožňují rychlý popis statistických veličin. Dále umožňuje pracovat s datovými typy jako jsou object, int64, float64, bool. Výhodou je i práce s indexy, které smí nabývat i časových hodnot. To umožňuje pracovat efektivně i s časovými řadami (Pandas, 2021).

2.2.2.4 NumPy

NumPy (dále jen NP) je open source projekt kompatibilní s programovacím jazykem Python. Ten je v této práci používán především pro práci s vícerozměrnými poli a maticemi. Právě pole jsou pro práci vzhledem k sekvenčním datům, jež slouží jako vstupní data do modelu LSTM, stěžejní (NumPy, 2021).

2.2.2.5 Matplotlib

Knihovny Matplotlib pomáhají s vizualizací dat. Umí pracovat s grafy jako jsou čárové grafy, histogramy, trojrozměrné grafy, sloupcové grafy, výsečové grafy, bodového grafy a jiné. Matplotlib bude používán pro grafické znázornění vstupů do modelu, aby došlo ke kontrole správnosti načtení dat. V neposlední řadě potom slouží jako grafické vyjádření

učení neuronových sítí, kde dochází k posuzování kvality modelu pomocí trénovacího a validačního souboru dat (Matplotlib, 2021).

2.2.3 Postup řešení úloh neuronových sítí

Veselý (2012, str. 26) radí řešit úlohy následujícím postupem:

1. Soubor dat je rozdělen na trénovací, validační a testovací
2. Je zvolena architektura sítě
3. Je zvolen učební algoritmus
4. Je proveden interaktivní proces učení. Proces učení končí, pokud je splněna ukončující podmínka.

V případě této práce je pracováno s časovou řadou, jež je rozdělena ze 70 % na trénovací z toho 20 % na validační a zbylých 30 % na testovací. Pro práci byla zvolena rekurentní síť, respektive model LSTM. U simulace dochází pak ke změnám konfigurace sítě, kde je například pracováno s různým počtem neuronů na vstupní vrstvě, a i ve vrstvách skrytých dále je pak přidána či ubrána jedna vrstva. Pro učení byl zvolen algoritmus zpětného šíření chyby (backpropagation). Podmínka pro předčasné ukončení, pokud se model nezlepšuje, je stanovena se zpožděním 30 interací.

3 Teoretická východiska

V teoretické části jsou vytyčeny základní poznatky umělé inteligence. Dále se pracuje s pojmy jako jsou časové řady a základní popisné statistické ukazatele, jež jsou pro práci s výběrovými soubory dat a vlastními výpočty použitých modelů stěžejní. V poslední podkapitole jsou vysvětleny základní metriky používané v e-commerce, které jsou přizpůsobeny v důsledku sjednocení názvosloví používané v inkriminované společnosti.

3.1 Umělá inteligence

Termín umělá inteligence se v posledních letech skloňuje čím dál tím častěji, a ne vždy je srozumitelné, co si pod ním představit. Je možné se setkat s mnoha definicemi, které tuto problematiku definují především z pohledu užití. Podle vyjádření Evropského parlamentu je umělá inteligence schopnost strojů se učit, plánovat, nebo napodobovat lidské chování (Evropský parlament, 2020). Profesor Veselý zmiňuje ve své publikaci „Úvod do umělé inteligence“ dva přístupy, a sice logický a neuronový přístup. Logický (též označovaný jako symbolický) tíhne spíše k vědnímu oboru psychologie a popisuje podstatu intelektu. V případě neuronového přístupu je umělá inteligence úzce spjata s neurofyziologií a neurobiologií (Veselý, 2005).

Pro účely této práce se tato kapitola zaměřuje především na obecné poznatky neuronu a neuronových sítí. Dále pak bude podrobně popsán specifický model, se kterým se dále ve vlastní část práce pracuje.

3.1.1 Historie umělé inteligence

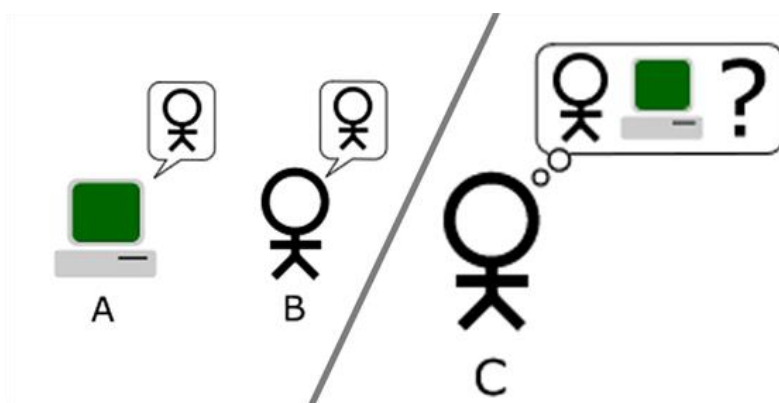
Počátky vzniku umělé inteligence se datují do poloviny dvacátého století. Především vznik návrhu matematického modelu neuronu, jež se stal pro neuronové sítě stěžejním. Tento objev mají na svědomí vědci McCulloch a Pitts, kteří rozvinuli teorii matematického modelu neuronu tak, jak ji známe dnes (Veselý, Úvod do umělé inteligence, 2005). V druhé polovině 20. století se vývoj této problematiky dostal do ústraní, neboť sklídl velkou kritiku především ze strany vědců Marvin L. Minského a Seymour S. Paret. Ti se domnívali, že nelze prostřednictvím neuronů provádět simulace logických zákonů. K obnovení výzkumu došlo až se založením Defence Advance Research Project Association, kde byl poprvé představen algoritmus Error Back-propagation of gradient, též známý pod názvem zpětné šíření chyby. Tento algoritmus se hojně využívá i v dnešní době (Tučková, 2009).

3.1.2 Turingův test a AI

V rámci historie vývoje umělé inteligence by neměl být opomenut zajímavý vědecký experiment, zvaný jako Turingův test. Autor, po němž je pojmenován, v roce 1950 zveřejnil esej, ve které se zamýšlí nad ověřením, zdali zkoumaný systém opravdu vykazuje inteligentní chování. Samotný průběh je poměrně jednoduchý.

V experimentu jsou přítomni dvě participující strany, které se nacházejí v samostatných oddělených místnostech, tak aby se navzájem nemohli ovlivňovat. V jedné místnosti je tazatel (člověk), a ve druhé místnosti je zařízení s „inteligentním“ programem (dále stroj) s dalším člověkem.

Obrázek 1: Turingův test



Zdroj: (Kasík, 2014)

Test probíhá tak, že tazatel klade různé otázky a druhá participující strana na ně odpovídá. Tazatel se následně snaží porozumět tomu s kým právě komunikuje. Zdali jde o stroj či člověka (Tvrdý, 2011). Samotný Turingův test je předmětem zkoumání a není obecně přijímaným postupem pro určení inteligentního systému.

3.1.3 Česko a AI

V České republice jsou ve výzkumu umělé inteligence nejvíce aktivní na Českém vysokém učení technickém, v Akademii věd ČR či Vysokém učení technickém v Brně.

Jednou z tváří, a především propagátorem umělé inteligence v České republice, je Sára Boutall. Ta stojí za zrodem organizace Prague AI, která si dala za cíl přilákat výzkumníky umělé inteligence ze zahraničí a pomoci novým talentům (Pohl, 2019). V mnoha rozhovorech hovoří o vhodnosti vývoje umělé inteligence v České republice. Řeší především pohled na vzdělání, jež je v této zemi velice kvalitní, a to především s ohledem na silný logicko-matematický základ (Koval, 2020). Prague AI však není jediným

sdužením, které se zaslouhuje o pozitivním přístupu k umělé inteligenci. Mezi dalšími jsou například Česko Digital, UNICO AI a jiné. Tyto společnosti spolupracují s firmami jako je Česká Spořitelna, Avast Software, CISCO, IBM, Microsoft a jiné, které podporují výzkum umělé inteligence v ČR.

V roce 2019 byla představen dokument NAIS, který jako první představuje národní strategii umělé inteligence v České republice mezi lety 2019 až 2030. Dokument pojednává o budoucí konkurenceschopnosti českých firem především v mezinárodním měřítku, ale zároveň navazuje na Koordinovaný plán Evropské komise (Ministerstvo průmyslu a obchodu, 2019).

3.1.4 Soft Computing

Široce užívaným pojmem, se kterým se lze v odborných literaturách setkat je Soft Computing (dále jen SC). Turing popisuje tradiční umělou inteligenci jako takovou simulaci inteligentního chování v systémech, které vyžadují přesnou a úplnou reprezentaci znalostí (Turing, 1950). S takto přesně určenými systémy se ale nelze v reálném prostředí setkat, ať už je to ovlivnitelností některých neočekávaných vlivů, které se těžko modelují. Mezi obory, které se těmito úvahami začali zabývat je Soft computing. Vznikl na konci 20. století. Za průkopníka v této oblasti je považován Lotfi A. Zadeh. Samotné „soft“ (neboli česky měkký) odkazuje na opak slova „hard“ (česky tvrdý), jež je přisuzováno tvrdým metodám výpočtů. V dalším uvažování je určitě vhodné si pár podobných slov, která daleko lépe vyjádří podstatu měkkosti výpočtů, sepsat do tabulky.

Tabulka 1: Tvrdé a měkké metody výpočtů

Tvrký	Měkký
Systematický	Nahodilý
Dobře definovaný	Vágní
Deterministický	Nedeterministický
...	...

Zdroj (Siddique & Adeli, 2013), vlastní zpracování

Z tabulky jasně vyplývá rozdíl mezi použitím metod hard, anebo soft computingu. Tedy na rozdíl od klasických tvrdých matematických metod se soft computing (dále jen SC) zabývá vágností, nahodilostí či nedeterminističností a to se promítá do užitých matematických modelů (Samir & Udit, 2013). SC tedy zahrnuje celou řadu systémů a to například: fuzzy systémy, neuro-fuzzy systémy, evoluční algoritmy, a i samotné neuronové

sítě (Veselý, 2005). Podle Volné et al. je hlavní podstatou SC specifikace pouze základních pravidel. Při rozhodování pak dochází k dedukci pravidel, podle nichž se dále řídí. Zároveň ale může docházet k nechtěným či dokonce neočekávaným chybám při některých výpočtech. To dokonce v krajních případech může vést až k újmě na životech osob (Volná, et al., 2014).

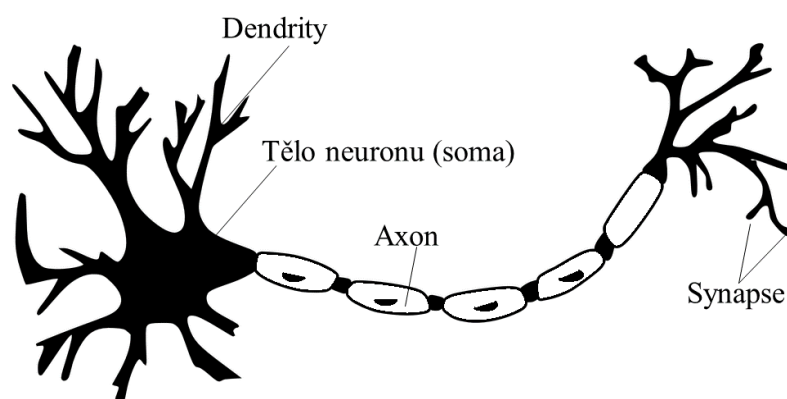
3.1.5 Neuron

Před tím než budou blíže popsány umělé neuronové sítě, není od věci podívat se na podstatu a fungování biologického neuronu, který je jednou z nejvíce záhadných oblastí vědeckého bádání. Některé studie uvádí, že se v mozku vyskytuje přes sto bilionů neuronů, jež jsou navzájem propojeny se svými sousedy (Samir & Udit, 2013). A i tak o přesné funkci všech procesů souvisejících s neurony víme tak málo. Co ale dokážeme popsat, jsou například jednotlivé procesy, které probíhají v určitých částech mozku, a i neuron samotný.

3.1.5.1 Biologický neuron

Biologický neuron je základní jednotkou nervové soustavy a slouží k přenosu signálů jak z interního, tak i z externího prostředí. Zjednodušeně se neuron popisuje prostřednictvím čtyř základních částí: tělo neuronu, synapse, axon a dendrity (Veselý, 2005). Každá z těchto částí má svůj specifický úkol.

Obrázek 2: Popis biologického neuronu



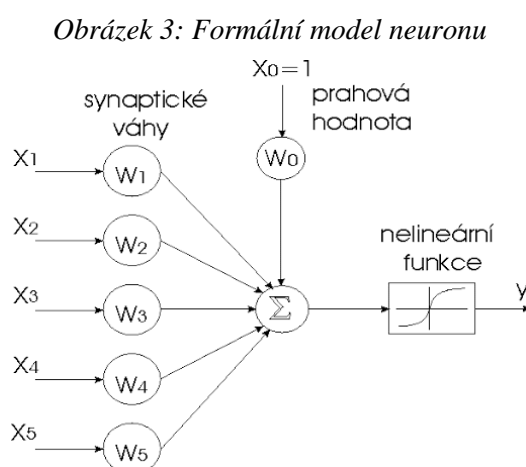
Zdroj: vlastní zpracování

Zjednodušený popis důležitých částí neuronu je zobrazen na obrázku 2. Dendrity jsou nervová vlákna přivádějící signál (vzruch) do buňky. Navazují přímo na tělo buňky a na ně se dále napojují synapse. Tělo neuronu (soma) je místem, kde se i mimo jiné vyskytuje jádro buňky. Zde dochází ke zpracování přijatého signálu. Od každého těla buňky vychází právě jeden axon, jehož délka může být i více než jeden metr. Axonem rozumíme nervové

vlákno, které spojuje sousedící nervové buňky. Dochází v něm k přeměně vstupujícího signálu na výstupní signál, který je následně dále veden směrem k synapsím. Synapse, nebo také synaptická spojení, jsou místem, kde dochází k vzájemnému propojení axonu s dendrity dalších neuronů. Pomocí nich je výstupní signál z jednoho neuronu veden jako vstupní signál do neuronu dalšího. A takto se zjednodušeně signál šíří z jednoho neuronu na další (Veselý, 2005).

3.1.5.2 Formální model neuronu,

Pokus o matematickou implementaci biologického neuronu proběhl již v roce 1943. McCulloch a Pitts představili tak zvaný formální neuron. Díky tomu byly položeny základy pro neuronové výpočty a později i neuronové sítě. Výpočty prováděné pomocí neuronových sítí se osvědčily zejména u velice složitých úloh, kde dochází k paradigmatu učení na základě příkladů. Z předešlého shrnutí biologického neuronu je možné sestavit základní poznatky pro vytvoření umělého neuronu. Formální neuron lze tedy rozdělit na 5 hlavních částí, a to vektor vstupů, synaptické váhy, práh, přechodovou funkci a výstup. Graficky může mít například následující zobrazení.



Zdroj: (Samir & Udit, 2013)

Vektor vstupů

Na samém počátku umělého neuronu je vektor vstupů X , který nabývá proměnných x_1 až x_n . Obdobně jako vstupuje vzruch pomocí dendritů do biologického neuronu.

$$x = (x_1, x_2, \dots, x_n); x_i \in R \quad (1)$$

Synaptické váhy

Synaptické váhy modelují reálné vlastnosti synapsí a udávají ohodnocení vstupů a jsou pro počet n prvků vyjádřeny reálným číslem. Podobně, jak tomu je například u

vícekriteriální analýzy variant, kde váhy vyjadřují důležitost jednotlivých kritérií, tak i zde jsou váhy vyjádřením důležitosti jednotlivých vstupů. Pomocí nich jsou potom vstupy přiřazovány na výstupy. Vektor vah W nabývá proměnných w_1 až w_n .

$$w = (w_1, w_2, \dots, w_n); w_i \in \mathbf{R}$$

Práh

Prahová hodnota vstupuje do neuronu jako tak zvaný fiktivní vstup. Potom tedy je tento vstup značen jako x_0 a pro něj přiřazená váha w_0 . Tato prahová hodnota v průběhu chodu algoritmu porovnává vstupní informace a rozhoduje o dalším přenosu. V některých literaturách se lze setkat také s pojmem bias. Obvykle se značí θ .

Potenciál

Oproti prahové hodnotě je postaven potenciál. Ten se vypočítá jako vážená suma vstupů vynásobených jejich synaptickými váhami. Značí se ξ . Porovnáním prahové hodnoty a potenciálu dále uvažujeme o výpočtu v přechodové funkci

$$\xi = \sum_{i=1}^n (w_i x_i - \theta), \quad (3)$$

kde w_i je vektor vah pro jednotlivé proměnné x_i .

Přechodová (aktivační) funkce

Přechodová funkce (v některých literaturách označována též jako aktivační funkce) se v tomto modelu především stará o jakousi transformaci vstupu na výstup neuronu (Tučková, Vybrané aplikace umělých neuronových sítí při zpracování signálů, 2009). Respektive rozhoduje o tom, zdali se signál z tohoto neuronu bude šířit dál. Je vyjádřena tvarem:

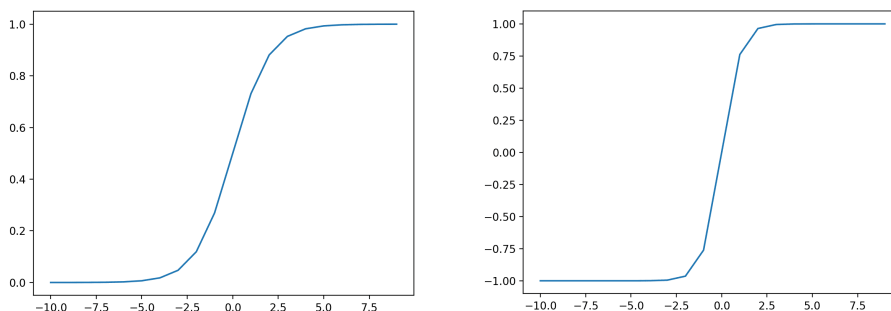
$$z = f(\xi), \quad (4)$$

kde ξ je potenciálem.

Aktivační funkce jsou jednou z hlavních součástí sítě. Výběrem specifické aktivační funkce je ve skrytých vrstvách ovlivňováno, jak dobře se model z dat učí. U výstupní vrstvy pak může ovlivňovat predikci modelu (Brownlee, 2019). Lze se setkat s nejčastěji používanými aktivačními funkcemi sigmoida, hyperbolická tangenta, inverzní tangenta, gaussova lineární funkce, skoková funkce či ReLU.

Sigmoidní aktivační funkce, také známá jako logistická, a hyperbolický tangens jsou zobrazeny na obrázku 4.

Obrázek 4: Aktivační funkce typu sigmoida (vlevo) a hyperbolický tangens (vpravo)

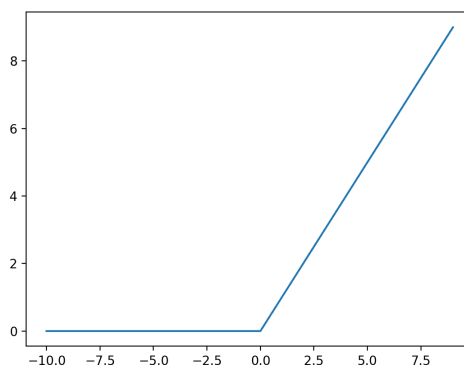


Zdroj: (Brownlee, 2019)

Pokud obě z funkcí berou v potaz vstupní i výstupní reálnou hodnotu. Tato hodnota se pohybuje pro sigmoidu v rozmezí 0 a 1. Pro hyperbolický tangens v rozmezí -1 a 1. Čím větší je vstup (kladný), tím se hodnota blíží 1, v opačném případě se blíží 0. U funkce hyperbolického tangentu se bude blížit -1 (Brownlee, How to use Learning Curves to Diagnose Machine Learning Model Performance, 2019).

Velice oblíbenou aktivační funkcí, především díky její jednoduché aplikaci, je rectified linear activation function (dále jen ReLU).

Obrázek 5: Aktivační funkce typu ReLU



Zdroj: (Brownlee, How to use Learning Curves to Diagnose Machine Learning Model Performance, 2019)

Ta funguje na jednoduchém principu. Pokud je hodnota na vstupu záporná, aktivační funkce vrátí hodnotu 0. V opačném případě je vrácena původní hodnota vstupu. Matematicky lze zapsat následovně:

$$\max(0, x) \tag{5}$$

Tato funkce řeší problém nasycení, kdy u velmi vysokých nebo velmi malých hodnot dochází v případě sigmoidy či hyperbolického tangentu ke sklouznutí vstupu k jejich krajním hodnotám 0, 1 nebo -1.

3.1.6 Učení

Stejně tak, jako je tomu i v reálném prostředí a kde je tato vlastnost nazývána plasticita synapsí, je i umělých neuronových sítí ono učení prováděno pomocí trénování souboru dat, kde dochází k přizpůsobování synaptických vah a prahů (Tučková, 2003).

Podle způsobu učení jsou neuronové sítě děleny na neasociativní, asociativní, jednorázové či s opakovaným učením. Asociativní učení vypovídá o schopnosti extrakce vztahů mezi jednotlivými stimuly. Neasociativní učení je zaměřeno na vybavování předem předložených stimulech a jeho následném vybavení (Biskup, 2009). Další jsou pak učení s učitelem a učení bez učitele. Podle portálu Helios se k *učení s* nebo *bez učitele* přidává i učení kombinované, které dokáže urychlit učení bez učitele a zpětnovazební učení, jež se používá především pro zpřesnění průmyslových strojů (Helios, 2017).

3.1.6.1 Učení s učitelem a bez učitele

Hlavní výhodou umělých neuronových sítí je forma učení. Nejčastěji je možné se setkat s učením s učitelem a bez učitele. V některých literaturách se nevyužívá českých překladů, ale zůstává se u anglických termínů supervised a unsupervised.

Rozdíl mezi těmito učitími typy je v trénovací množině. U učení s učitelem se pracuje jednak s hodnotami na vstupech, tak i s požadovanými hodnotami na výstupech. Ty se potom v průběhu výpočtů porovnávají s aktuálním výstupem ze sítě. Jinými slovy učení není nic jiného než minimalizace chybové funkce. Pro každý vzor v testovací množině platí minimalizace rozdílů mezi požadovanou hodnotou a výstupní hodnotou. Vyjádřeno rovnicí jde o minimum součtu rozdílů hodnot y_k a d_k , kde y_k je výstupní hodnota a d_k je považovaná hodnota.

$$\min(E^t) = \min \sum_{k=1}^s (y_k - d_k)^2 \quad (6)$$

Tučková proces tohoto učení přirovnává k optimalizačnímu procesu, kde dochází k optimalizaci účelové funkce ve formě chybové funkce. V tomto případě ale ÚF symbolizuje chybovost daného modelu při učení s učitelem. V učení tedy dochází k modifikaci synaptických vah a prahů (Tučková, 2009). Tento druh učení se vyskytuje u sítí Perceptron,

Adaline (adaptivní lineární neuron), Medaline a vícevrstvé sítě se zpětným šířením chyby (Backpropagation) (Tučková, 2003).

3.1.6.2 Učení bez učitele

Učení bez učitele je využíváno u úloh, které se primárně soustředí na hledání podobností ve vstupních datech. Díky neuronové síti je učení schopné třídit data do tak zvaných shluků (clusters). Clustering je právě jednou z používaných technik učení bez učitele a v reálném životě se používá například u segmentace zákazníků (Yagcioglu, 2020).

3.1.7 Neuronové sítě

Po znovuoobnovení vývoje umělé inteligence v osmdesátých letech minulého století, se začalo daleko důkladněji bádát v oblastech uskupení neuronů. V lidském těle se vyskytuje 10^9 neuronů, které mezi sebou umožňují prostřednictvím synapsí přenášet signál (Veselý, 2005). Jejich propojením se položily základy neuronovým sítím a jejich následnou modifikací na reálné praktické příklady došlo k velice pozitivním výsledkům. Neuronová síť tedy obecně vzniká spojením jednotlivých neuronů. Výstupy z neuronu prvního se stávají vstupy do neuronů dalšího, někdy i do více neuronů. Potom dochází k tvorbě sítě neuronů.

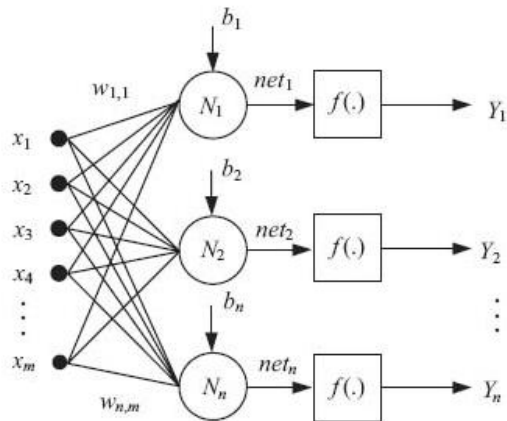
3.1.7.1 Architektura neuronových sítí

Architekturu vícevrstevných neuronových sítí lze rozdělit do dvou základních skupin podle způsobu propojení jednotlivých neuronů ve vrstvách, jimiž jsou dopředné sítě a rekurentní sítě. Topologie sítí značně ovlivňuje výpočty.

Dopředné sítě

Jednoduchým příkladem dopředné sítě může být dopředná síť s jednou vrstvou, jež je znázorněna na obrázku Obrázek 6.

Obrázek 6: Dopředná síť s jednou vrstvou



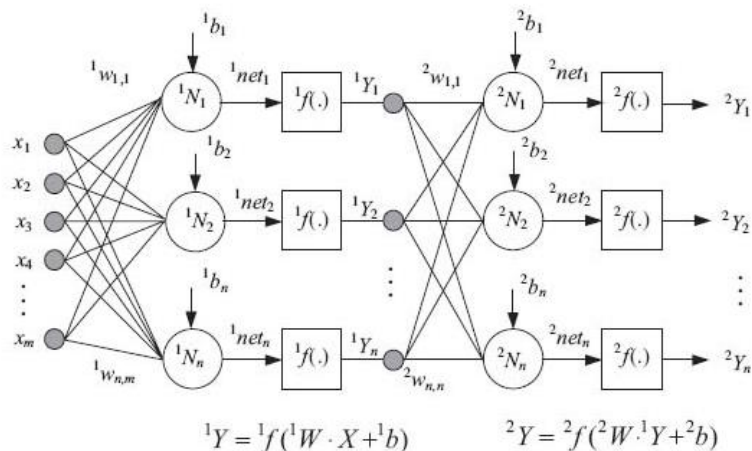
Zdroj: (Samir & Udit, 2013)

Takto například vypadá síť, která se skládá z více neuronů v jedné vrstvě. Jedná se o síť *Single-layer feedforward network*. Neurony jsou znázorněny pod sebou a označeny velkými písmeny N. Do každého neuronu vstupují neurony x_m s váhami w_{nm} . Nejedná se o více vstupních vektorů, nýbrž pouze o jeden, jehož jednotlivé proměnné vstupují do každého z neuronů zvlášť. Mohou mít ale různé váhy. Do každého neuronu také vstupuje samostatný bias b_i . Množství biasů je závislé na počtu neuronů v síti. Výstupem této sítě je vektor výstupů Y pro jednotlivé neurony N_1 až N_n .

Dopředná síť s více vrstvami

Pokud se struktura sítě rozšíří o jednu řadu neuronů, pak mluvíme o dopředné síti s více vrstvami. Tyto další vrstvy, které se nacházejí mezi vstupní a výstupní vrstvou nazýváme skryté vrstvy (hidden layers). Těch může být mnoho podle potřeby výpočtů a použitých konkrétních typů sítě. Příkladem může být síť na obrázku 7.

Obrázek 7: Dopředná síť s více vrstvami

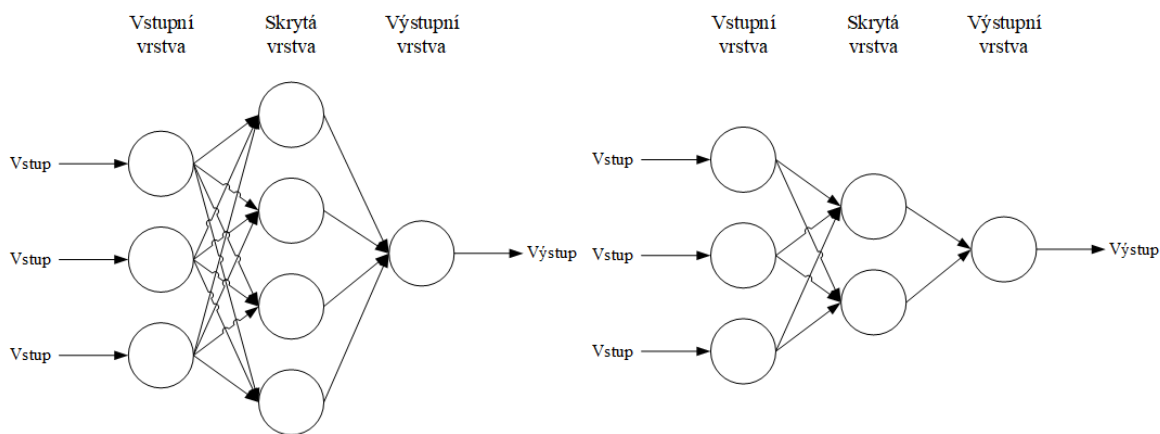


Zdroj: (Samir & Udit, 2013)

Způsob průběhu je podobný jako u jednovrstvé sítě. Důraz je potom kladen na výstupní hodnoty z první vrstvy 1Y_n , které následně vstupují do vrstvy další, s neurony 2N_n , a tento proces se opakuje tak dlouho, kolik se nachází skrytých vrstev v síti (Samir & Udit, 2013). Vstupům jsou zároveň přiřazeny váhy 2w_n .

Podle počtu neuronů v jednotlivých vrstvách se dá podle Tučkové topologicky rozdělit síť na dva typy. První má pyramidální tvar, kdy každá následující vrstva je minimálně o jeden neuron chudší a takto se pokračuje až po výstupní neuron. Druhý typ je síť, která má vstupní vrstvu menší než skrytou. Například na obrázku 8 je znázorněna síť, která je na druhé vrstvě, respektive první skryté, rozšířena (Tučková, 2009).

Obrázek 8: Síť s rozšířenou skrytou vrstvou (vlevo) a síť pyramidální (vpravo)

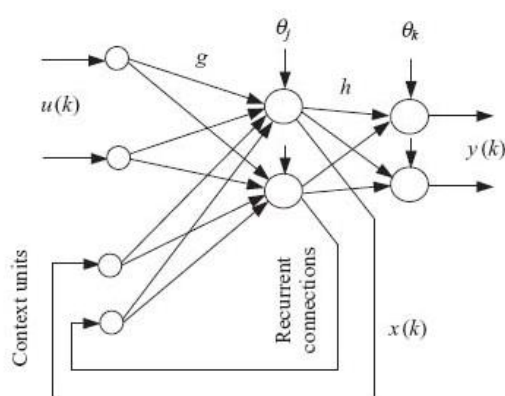


Zdroj: vlastní zpracování

3.1.8 Rekurentní neuronové sítě

U dopředných sítí se signál šíří pouze jedním směrem – dopředu a neumožňuje se tak vracet či točit ve smyčce. Toto však dokážou sítě rekurentní (Materna, 2019). Ty jsou považovány za daleko složitější, ale daleko více se přibližují způsobu fungování lidského mozku. Rekurentní neuronové sítě (dále jen RNN) nejen že jsou schopny se vracet, ale dokážou si zapamatovat data z předešlých kroků výpočtů. RNN tedy obsahují paměť, se kterou dále podle specifik zvoleného modelu pracují. To je užitečné v případech závislosti dat na předchozích hodnotách, například v časových řadách (Tučková, 2009). Stejně jako u dopředných sítí je i u těchto typů sítí mnoho druhů. Mezi nejjednodušší patří Elmanova síť.

Obrázek 9: Ukázka rekurentní neuronové sítě



Zdroj: (Samir & Udit, 2013)

Tato síť používá algoritmu zpětného šíření chyby. K základní neuronové síti je přidána kontextová vrstva sítě. Ta umožňuje vstup výstupních údajů z předešlé vrstvy do stejné vrstvy jako vstupují další vstupní data.

Takto jednoduchý model je pro potřeby této práce nedostačující. Daleko kvalitnějším RNN modelem je Long-short term memory či jeho jednodušší modifikace Gated recurrent unit.

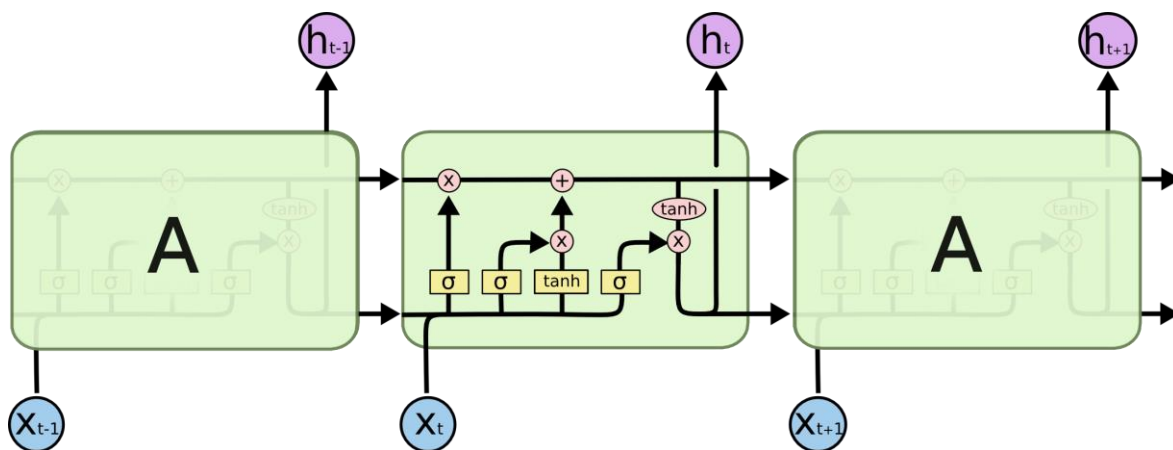
3.1.9 Model Long-short term memory

Model Long-short term memory (dále jen LSTM) je obdobou konvolenčních neuronových sítí. Díky vlastnosti rekurentních neuronových sítí, mezi něž se LSTM také řadí, je hojně využíván v oblastech, jako jsou rozpoznávání mluveného slova či písma, nebo při práci s časovými řadami jež jsou závislé na předchozích hodnotách. Jinými slovy tam, kde je nutné zapamatování dlouhé časové periody (Olah, 2015). Tento algoritmus byl

představen již v roce 1997. Za jeho hlavní představitele se považují Hochreiter a Schmidhuber. Hlavním rozdílem od běžné RNN je fakt, že ve výpočtech běžných RNN dochází k tak zvanému problému mizejícímu gradientu (vanishing gradient) (Arbel, 2018). Tento problém je vázaný na konvergování gradientu v dlouhém časovém horizontu k nule. Tomu bylo u této rekurentní sítě zamezeno přidáním LSTM paměťových buněk.

Na obrázku 10 je grafické znázornění modelu LSTM. Černé šipky popisují tok vektorů dat v buňce. Růžové kruhy pak signalizují operace. Žlutá políčka pak vypovídají o naučených neuronových sítích. V případě rozdvojení se jedná o proces kopírování. V opačném případě pro slučování jde o proces zřetězení (Olah, 2015).

Obrázek 10: Schéma modelu LSTM

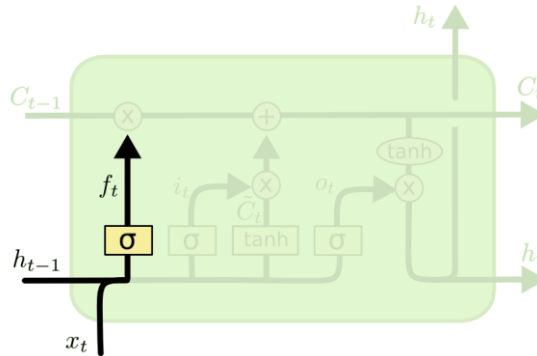


Zdroj: (Olah, 2015)

Průběh algoritmu LSTM

Postup využití LSTM lze zjednodušeně popsat v několika úkonech. Nejprve je učinit rozhodnutí, jaké informace vyhodíme z cell state (vodorovná šipka procházející celou buňkou), což je pak následně provedeno pomocí sigmoidní vrstvy, které je označovaná jako „forget gate layer“. Model se podívá na h_{t-1} a x_t a pro každé číslo v cell state C_{t-1} vytvoří výstup v intervalu 0 a 1, kdy zjednodušeně 0 znamená „tohoto je nutné se zbavit“ a 1 naopak „toto je potřeba zachovat“. To znázorňuje i následující obrázek 11.

Obrázek 11: Forget gate layer



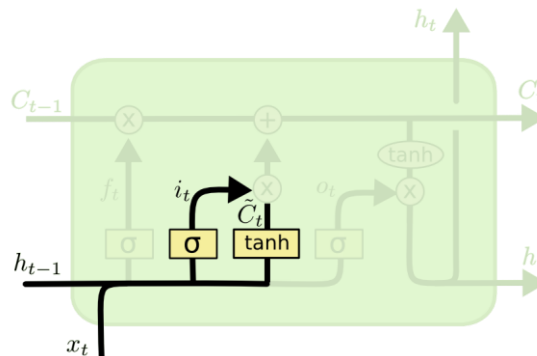
Zdroj: (Olah, 2015)

Matematicky lze vyjádřit takto:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (7)$$

Dále je nutné udělat další rozhodnutí, jaké nové informace budeme ukládat do cell state, které spočívá ve dvou krocích. Nejdříve zmíněná sigmoidní vrstva rozhodne, ze kterých hodnot bude vytvořen výstup. Poté vrstva tanh přidělí vektor kandidátům na nové hodnoty C_t , které je možné přidat do state. Kombinace těchto dvou úkonů vytvoří aktualizaci state. To umožní provést aktualizaci staré cell state C_{t-1} na nový C_t .

Obrázek 12: Proces ukládání informací do cell state



Zdroj: (Olah, 2015)

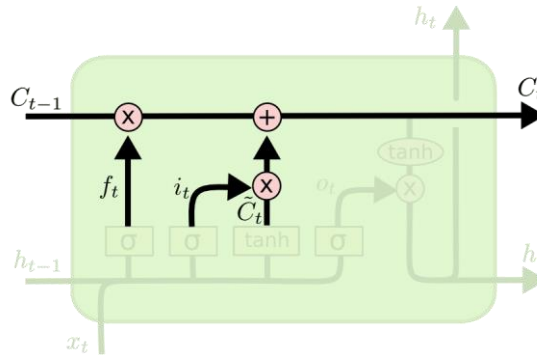
Matematicky lze tento první krok vyjádřit následovně:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (9)$$

Věci, které byly označeny, že mají být zapomenuty, jsou zapomenuty a násobíme starý state funkcí f_t . Poté se přidá $i_t * C_t$, což jsou kandidáti na nové hodnoty, škálované podle rozhodnutí, jak moc jsme chtěli každou state hodnotu.

Obrázek 13: Zápis hodnot do state



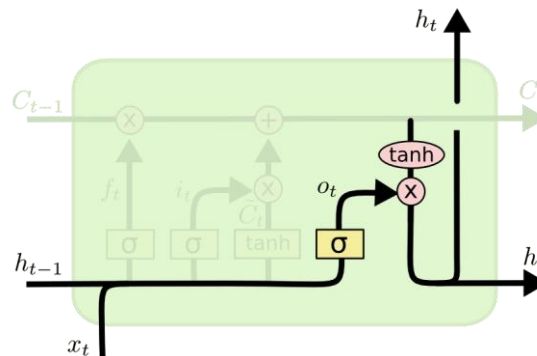
Zdroj: (Olah, 2015)

Zápis tohoto kroku je následující:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (10)$$

Na závěr se musíme rozhodnout, co má být výstupem, neboť ten bude sice založen na našem cell state, ale bude to jeho jakási filtrovaná verze. Proto nejprve spustíme sigmoidní vrstvu, jež učiní rozhodnutí, jaké části cell state půjdou do výstupu. Pak vložíme cell state do tanh, aby se hodnoty dostaly do intervalu mezi -1 a 1 a nakonec je vynásobíme výstupem ze sigmoidní brány tak, abychom získali pouze výstup částí, které chceme.

Obrázek 14: Výstup z buňky



Zdroj: (Olah, 2015)

Matematicky vyjádřeno takto:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

Takto funguje základní LSTM model. Samozřejmě existuje celá řada obdobných modifikací tohoto algoritmu. Jedním z nich je například i GRU (Gated Recurrent Unit), který v roce 2014 představil Kyunghyun Cho. Ten byl zjednodušen ubráním jedné brány a díky tomu oproti LSTM dochází k menšímu zatížení při výpočtech, respektive průběh testování

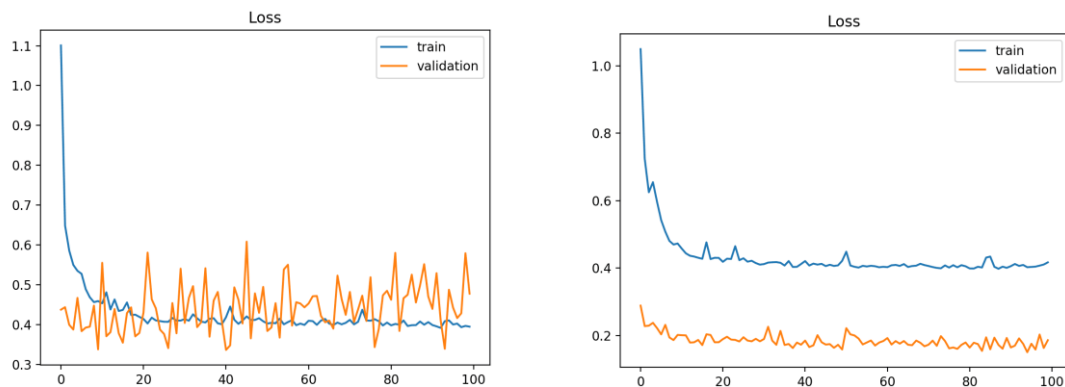
souboru není tak náročný. V obou případech modely vykazují podobné výsledky (Cho, et al., 2014).

3.1.10 Posouzení kvality modelu

Pro vyhodnocování kvality modelu není použití běžných statistických parametrů stěžejní. Je vhodné zobrazení grafů, kde se posuzují validační a testovací křivky. Trénovací křivka vypovídá o tom, jak dobře se model učí a validační křivka vypovídá o tom, jak dobře model zobecňuje. Podle jejich průběhu je možné usuzovat tři různé scénáře (TriloByte, 2021).

1. Špatná predikční schopnost – validační křivka je pod křivkou trénovací nebo vykazuje zcela jiný průběh

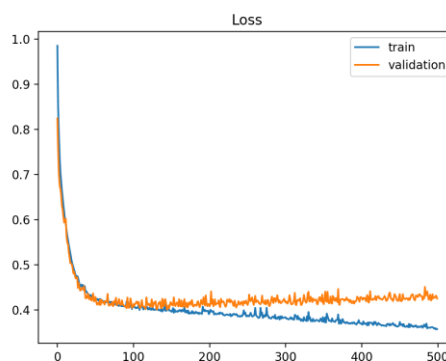
Graf 1: Špatná predikční schopnost



Zdroj: (Brownlee, 2019)

2. Uspokojivá predikční schopnost – validační křivka je nad křivkou trénovací.

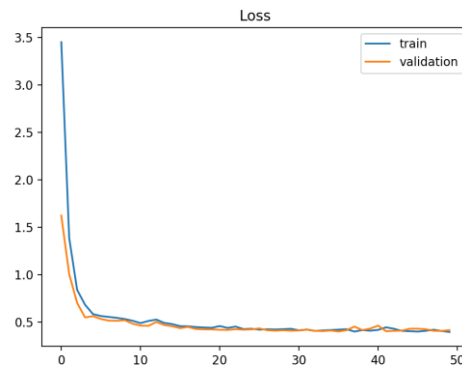
Graf 2: Uspokojivá predikční schopnost



Zdroj: (Brownlee, 2019)

3. Velmi dobrá predikční schopnost – validační a trénovací křivky se překrývají

Graf 3: Velmi dobrá predikční schopnost

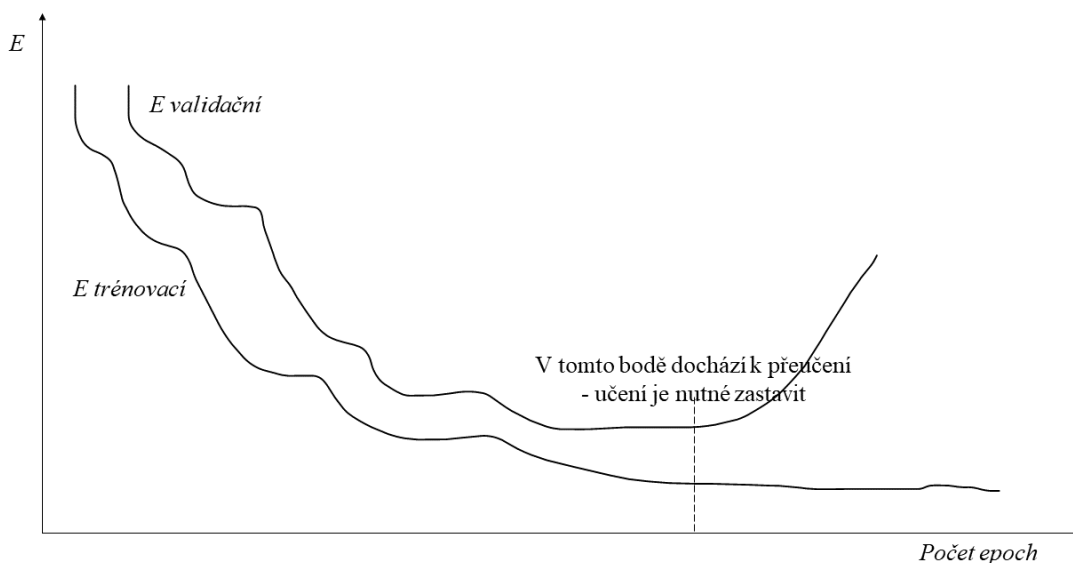


Zdroj: (Brownlee, 2019)

3.1.11 Přeučení sítě

Pokud po určitém počtu epoch dochází ke zvyšování chyby u validačního souboru, tak dochází k tak zvanému přeučení. V tomto případě síť ztrácí schopnost generalizace. Generalizací je myšlena schopnost sítě, kdy dokáže na základě naučeného trénovacího souboru usuzovat na nové případy, aniž by je trénovací soubor obsahoval. V tento moment je nutné učení zastavit (Veselý, 2005).

Obrázek 15: Přeučení sítě



Zdroj: (Veselý, 2012)

3.2 Základní metriky používané v e-commerce

V prostředí e-commerce se využívá celá řada různých metrik. Zmíněny jsou jen ty, které jsou relevantní pro tuto diplomovou práci. Z hlediska vstupních dat do modelu je pracováno především s tržbami, transakcemi a Average Order Value (dále jen AOV).

3.2.1 Tržby

Tržby se v prostředí e-commerce běžně označují jako „revenue“. Obecně platí, že tržby neobsahují veškeré hodnoty transakcí. Google Analytics 360 nedokáže v případě zkušebních licencí počítat s přepočty hodnot za aktivaci u uživatelů využívajících trialového období, proto se některé licence dopočítávají ručně. To lze vyjádřit vztahem:

$$\text{Celkové tržby} = \text{Přímé tržby} + \text{Aktivace trialových licencí} * \text{AOV} * \text{CR},$$

kde aktivace zkušebních licencí je vynásobena AOV a CR (Conversion Rate, v tomto případě vyjadřuje možnost aktivace jiných kupónů po vypršení zkušební doby). CR je konstantou, jež se jen zřídka mění. V této práci je pracováno s celkovými tržbami, které se budou dále nazývat jenom tržby.

3.2.2 Transakce

Transakce jsou vyjádřeny počtem přímých nákupů. Výpočet celkových transakcí je obdobný jako v případě tržeb. V jiných případech jsou Transakce složité na výpočet a jsou vázány na session potencionálního kupujícího.

$$\text{Celkové transakce} = \text{Přímé transakce} + \text{Aktivace trialových licencí} * CR$$

3.2.3 Average Order Value

Průměrná hodnota objednávky (AOV) je metrika elektronického obchodování, která sleduje průměrnou částku dolaru utracenou pokaždé, když zákazník zadá objednávku na webu nebo v aplikaci. AOV je považována za jednu z nejdůležitějších metrik v odvětví elektronického obchodování (Contants, 2018).

AOV je ale především ukazatel, který se právě používá k určení, jaké si výkonosti kupónu produktu. Toho se dá využít například při tvoření strategie cenových plánů. Dále se využívá jako jedna z hlavních metrik při testování změn cen například v AB testování, kde v případě varianty došlo k pozměnění ceny na jednotlivých kupónech. Obecně vzato to znamená, že se podle AOV stanovuje například zda kupón posiluje, respektive je žádanější, či naopak je dobré kupón zcela stáhnout z nabídky a dále jej nenabízet.

$$AOV = \frac{\text{Tržby}}{\text{Transakce}} \quad (13)$$

AOV se vypočítá jako podíl celkových tržeb a celkových transakcí. Obdobou AOV jsou další používané metriky, například PSV (Per Session Value), RPM (Revenue Per Thousand Impressions).

3.3 Časové řady

Pokud má být dosaženo kvalitních výsledků, musí být pracováno s obsáhlými datovými sadami, což však někdy může být obtížné. Ne vždy lze některá data získat, anebo je dat naopak celá řada, ale bylo s nimi špatně zacházeno a v širším pohledu nedávají smysl či jsou dokonce poškozena.

Časovou řadu lze definovat jako výskyt pozorování v čase. Mezi jednotlivými údaji souboru jsou pozorovány stejné časové intervaly a jsou zároveň chronologicky uspořádány. Tyto řady jsou zatíženy stochasticitou, čili určitou nahodilostí (Křivý, 2012).

3.3.1 Základní charakteristiky

Mezi základní charakteristiky rozdělení souboru patří aritmetický průměr, rozptyl a medián. Jejich interpretací lze dosáhnout obecných poznatků o chování souboru dat ve sledovaném období.

Aritmetický průměr

$$\bar{Y} = \frac{1}{n} \sum_{t=1}^n Y_t \quad (14)$$

Rozptyl

Rozptyl vypovídá o rozptýlení skutečných (pozorovaných) hodnot od průměrné (vypočtené) hodnoty.

$$S_Y^2 = \frac{1}{n} \sum_{t=1}^n (Y_t - \bar{Y})^2 \quad (15)$$

Směrodatná odchylka

Obdobně jako v případě rozptylu směrodatná odchylka určuje odchýlení hodnot od průměru a je druhou odmocninou rozptylu.

$$S_Y = \sqrt{S_Y^2} \quad (16)$$

Medián

Výpočet mediánu je závislý na sudosti či lichosti výběrového souboru dat. Pokud je sudý, lze výpočet provést následujícím algoritmem:

1. Seřazení hodnot od nejnižší hodnoty po nejvyšší
2. Nalezení prostředních dvou hodnot souboru dat
3. Zprůměrování těchto hodnot

Pokud je výběrový soubor dal lichý v počtu dat, poslední krok odpadá, neboť je prostřední hodnota pouze jedna.

3.3.2 Regresní modely

Popisovat regresní modely je na samostatnou práci ale malý úvod si, i vzhledem k dalšímu používání tohoto termínu, zaslouží. Co to tedy regresní model je? Je to model, který se snaží aproximovat daná pozorování. Tato aproximace je ovlivňována použitým funkčním předpisem regresní funkce. Mezi nejznámější patří lineární regresní model, ten je brán i jako nejjednodušší. Jeho základem je, jak již z názvu vyplývá, předpis lineární funkce.

$$Y_i = \alpha + \beta X_i + \varepsilon_i, i = 1 \dots n, \quad (17)$$

kde Y je závisle proměnnou a X nezávisle proměnnou. Náhodná chyba je značena ε pro hodnot $i = 1 \dots n$. α a β jsou odhadované parametry pomocí metody nejmenších čtverců. Mimo lineární regresní model existují i vícenásobné (někdy také mnohonásobné) modely (Zichová, 2021).

3.3.3 Regrese a neuronové sítě

Neuronová síť je jakousi vícenásobnou nelineární regresí. V případě neuronové sítě bez skryté vrstvy, kde je pouze vstupní a výstupní vrstva se jedná o lineární regresní funkci. Síť se potom sama optimalizuje v závislosti na minimalizaci rozdílů predikovaných a skutečných hodnot. Optimalizace je v tomto duchu možná až moc silné slovo. Dá se předpokládat, že optima ve výpočtech nelze dosáhnout. Lze se k němu pouze přibližovat a uzpůsobovat modely tak, aby se snižovala chyba předpovědi. Oproti běžným regresním modelům dokáže neuronové sítě vysvětlovat s daleko menším rozdílem od skutečných hodnot původní funkce. Respektive tyto sítě aproximují daleko lépe původní funkci. To je jednou z největších výhodou při používání regresních neuronových sítí.

3.3.4 Chyba předpovědi

Chybu předpovědi se běžně označuje e_t . Stanovuje se jako rozdíl mezi skutečnou (naměřenou) hodnotou Y_t a hodnotou předpovědi $Y_t(t-1)$ (Křivý, 2012).

$$e_t = Y_t - Y_t(t-1) \quad (18)$$

Samotná jedna chyba určité předpovědi u časové řady obsahující mnoho pozorování v čase t je nedostačující, a proto se používají metriky pro určení míry kvality předpovědi. Mezi základní z nich patří SSE (Sum of Squared Errors), MSE (Mean Squared Error), MAD (Mean Absolute Deviation), MAPE (Mean Absolute Percentual Error).

SSE je tedy dáno součtem chyb umocněn na druhou, tedy

$$SSE = \sum_{t=1}^n e^2 \quad (19)$$

Při posuzování rozdílu mezi predikovanými a skutečně pozorovanými hodnotami se v modelech hlubokého učení, respektive u modelů regresních neuronových sítí užívá RMSE (Root-mean-square deviation).

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_t - \hat{x}_t)^2}{N}}, \quad (20)$$

kde se ve své podstatě jedná o podíl SSE a počtu pozorování. To celé je dále odmocněno. To znamená, že RMSD je ve své podstatě směrodatná odchylka reziduí. Výsledná hodnota vypovídá o tom, jak jsou data kolem regresní funkce koncentrována (Barnston, 1992).

4 Vlastní práce

4.1 Základní analýza dat

Pro správné pochopení dalších úvah o takto specifickém problému je nutné provést důkladnou analýzu potřebných dat, které dále poslouží jako vstupní data do modelu umělých neuronových sítí.

4.1.1 Získání dat

Pro získání dat bylo využito především internetového rozhraní Google Analytics 360. Velkou výhodou, jak již bylo zmíněno v metodické části, je možnost napojení Microsoft Power BI přímo na databázové rozhraní GA360. To však neumožňuje pracovat se segmenty a nelze tak zcela čistě vyselektovat některá z dat. V neposlední řadě bylo pracováno i s Microsoft Office 365 – Excel, kde docházelo pouze k úpravě souborů typu csv. V některých případech se získávaly potřebné informace prostřednictvím výstupu z programovacího jazyka Python, respektive knihoven Pandas za použití programu Jupyter. Prostřednictvím specifických knihoven se z tohoto prostředku stal vhodný nástroj pro okamžité zobrazení potřebných deskriptivních statistických popisů souborů dat a pro případnou kontrolu zdali je soubor, respektive jednotlivé sloupce, v požadovaném formátu. Prostřednictvím softwaru PowerBI byly porovnány rozdíly mezi výpočty. Vzhledem k možnostem automatizace některých výpočtů se tak urychlila práce se správou značného množství dat najednou.

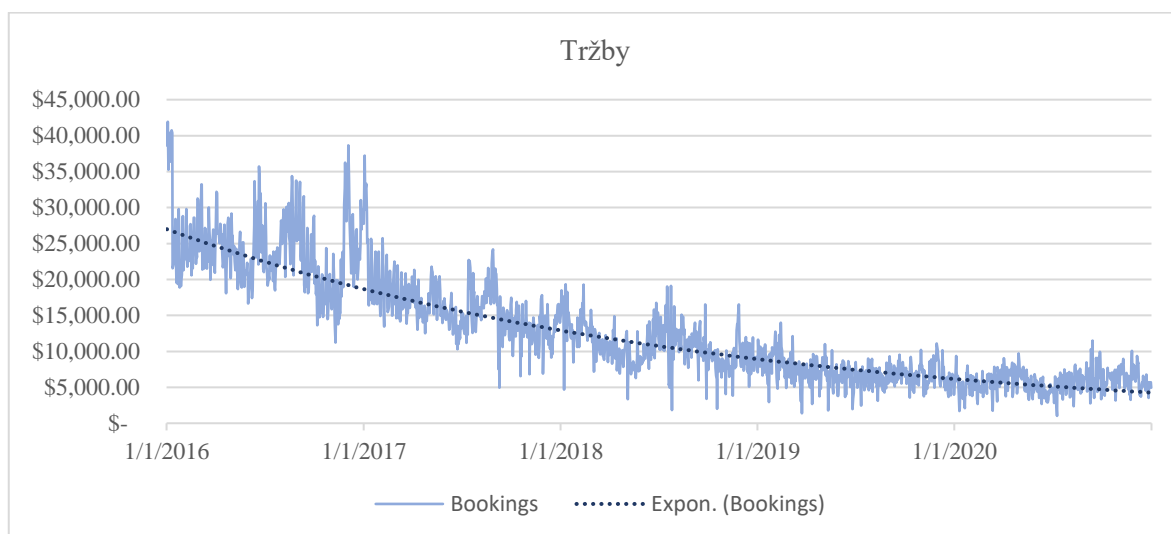
4.1.2 Základní analýza dat

Z již předešlých poznatků z části metodické a teoretické, je souhrn těchto dat stanoven na přesně definovanou zemi a cenu kupónu, za který je elektronická licence kupována. Mohlo by se zdát, že se jedná o poměrně fixní cenu, která se v průběhu času nemění, nebo jen velmi nepatrně. Avšak k modifikaci dochází v případě zapojení dalších objektů podílejících se na výsledném AOV. Mezi takové patří právě oddělení podpory, kde dochází k nejvýraznějším výkyvům od průměrné ceny. V některých situacích dochází i k aktivitám, kdy jsou kupóny prodávány silně pod průměrnou cenou. O tom ostatně vypovídají i následující analýzy souborů dat.

4.1.2.1 Tržby

Na následujícím grafu se nacházejí celkové tržby v čase.

Graf 4: Tržby v letech 2016–2020

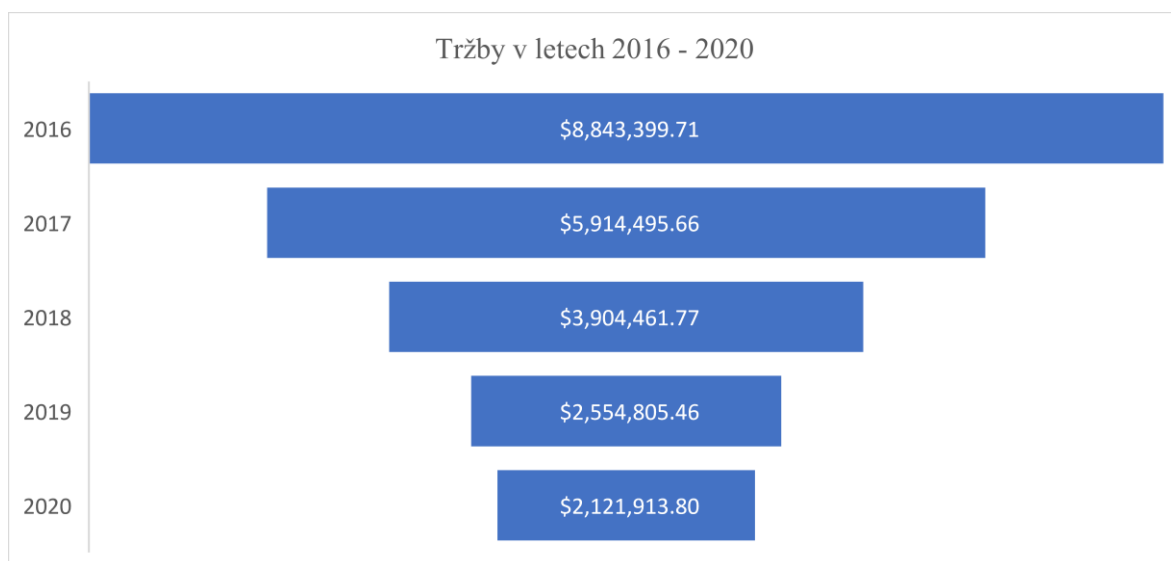


Zdroj: vlastní zpracování

Z grafu je pozorovatelný dlouhodobý pokles, který se dá přirovnat k exponenciální funkci. V letech 2019 dochází ke stagnaci a následně, v roce 2020, už je patrný malý nárůst, obzvláště v posledních měsících. U mezikvartálního porovnání dochází v roce 2019 k poklesu přibližně o 1 % v průměru, zato v roce 2020 v průměru o 4 % nárůst. O tom ostatně, při pohledu na grafické vyjádření, svědčí i rostoucí vzdálenost od exponenciály u sledovaných hodnot.

Další zjištění jsou patrná z následujícího grafu srovnání tržeb podle jednotlivých let.

Graf 5: Sumarizace tržeb v jednotlivých letech

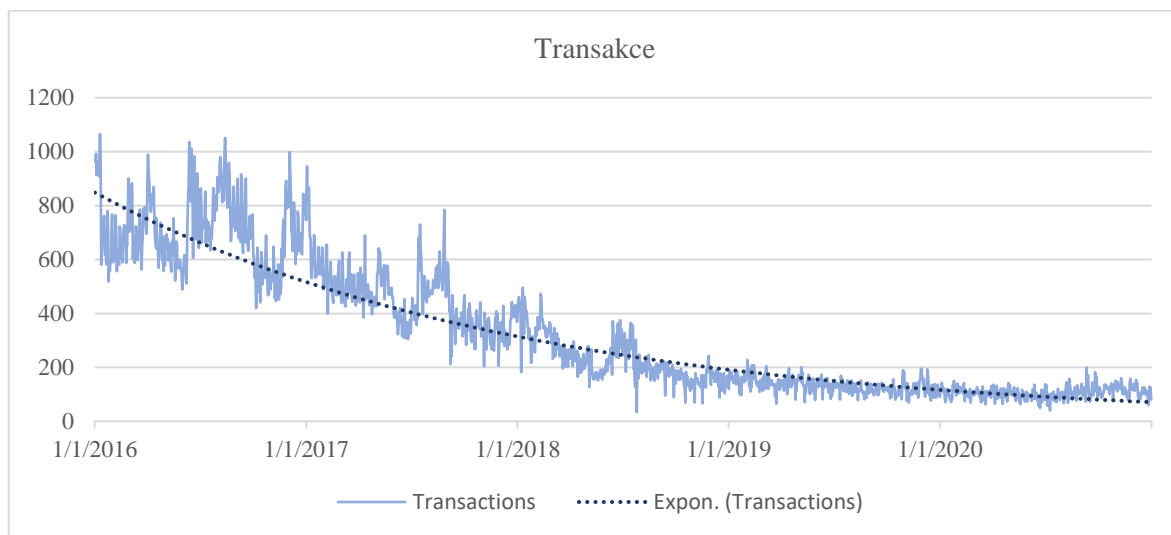


Zdroj: vlastní zpracování

Z grafu je patrný pokles celkových tržeb oproti roku 2016 až o 70 %. Největší meziroční pokles byl pak sledován mezi obdobími 2018 a 2019 a to přibližně o 45 %. Nejmenší pak v letech 2019 a 2020 a to o 17 %. Lze tedy i zde uvazovat nad zpomalováním trendu poklesu celkových tržeb.

4.1.2.2 Transakce

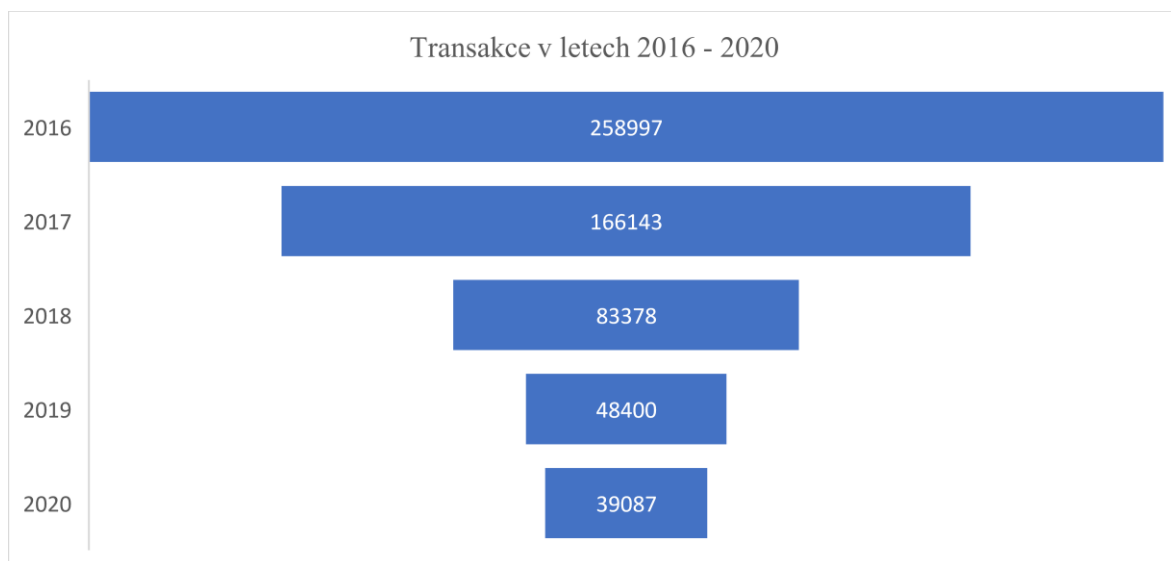
Graf 6: Transakce (ks) v letech 2016 - 2020



Zdroj: vlastní zpracování

Stejně tak jako tomu bylo u tržeb, transakce exponenciálně klesaly a na konci roku 2020 je opět sledován malý nárůst. V mezikvartálním srovnání pak v průměru dochází v roce 2019 k poklesu o přibližně 7 %, oproti tomu v roce 2020 zase k nárůstu o přibližně 8 %.

Graf 7: Sumarizace transakcí (ks) za jednotlivé roky



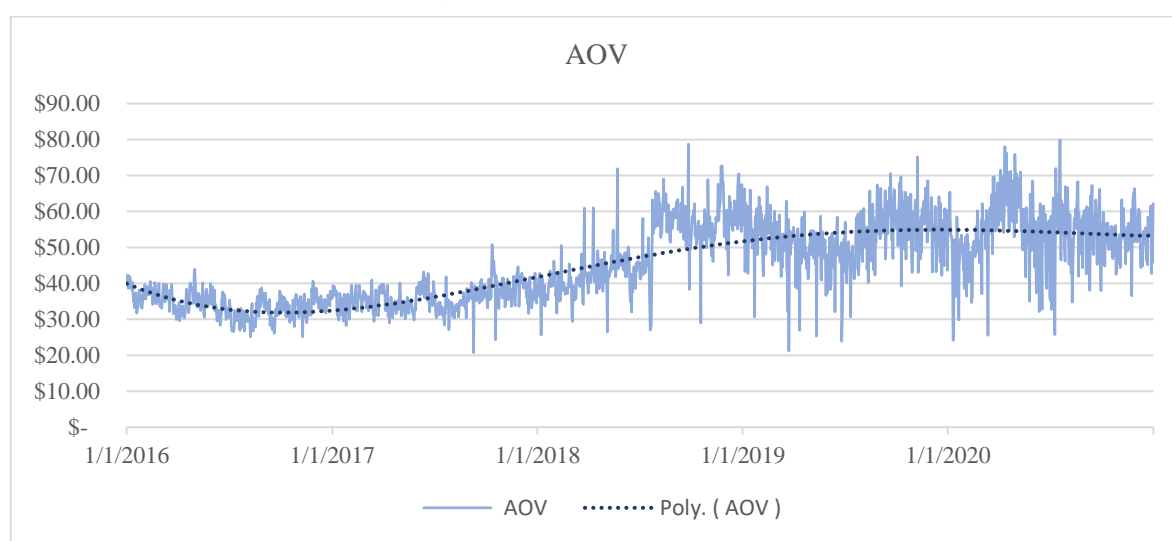
Zdroj: vlastní zpracování

Podobně jako u tržeb dochází ke snižování počtu transakcí. Nejvýraznější pokles je meziročně u let 2017 a 2018 a to přibližně o 50 %. Nejmenší meziroční pokles je pak mezi obdobími 2019 a 2020, a to o 19 %. Rozdílem let 2016 a 2020 je procentuální snížení daleko vyšší než v případě tržeb. To je rovno -85 %.

4.1.2.3 AOV

Výpočtem podílu tržeb a transakcí je vypočteno AOV. To vypadá následovně.

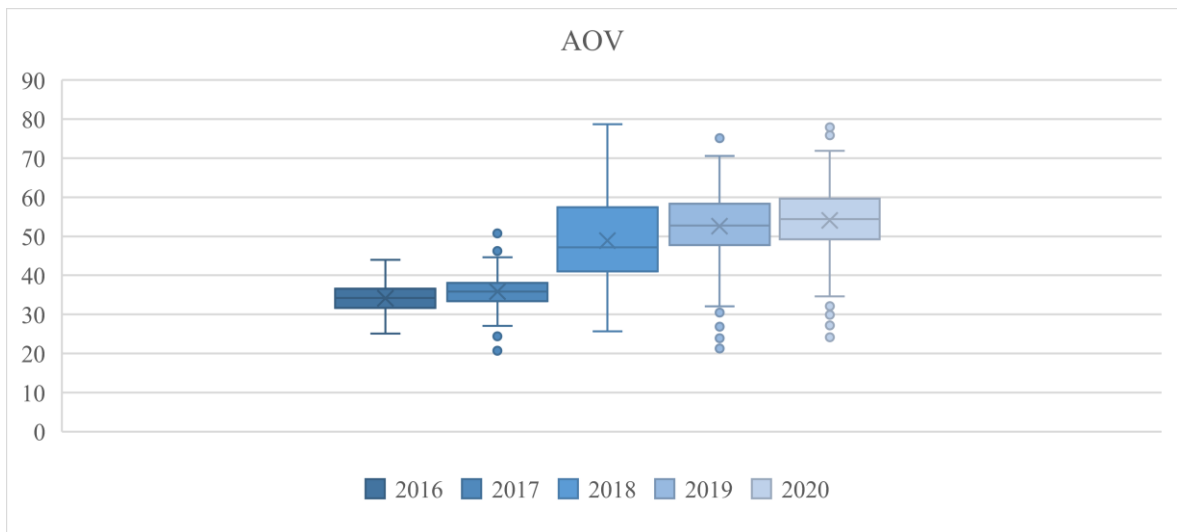
Graf 8: AOV v letech 2016 - 2020



Zdroj: vlastní zpracování

Pokud bude sledována průměrná hodnota AOV v jednotlivých letech, pak minimální hodnotou je 34,15 USD v roce 2016. Maximální průměrná hodnota je pak v roce 2020 a to 54,04 USD to ale může být ovlivněno nalezením maximální hodnoty právě v tomto období. Všechny hodnoty AOV se nacházejí v intervalu 20,70 USD spodní hranice a 79,91 USD horní hranice. Podrobnější znázornění rozdělení souboru je vizualizováno na grafu boxplotů.

Graf 9: Přehled boxplot grafů projednotlivé roky



Zdroj: vlastní zpracování

V grafu výše je znázorněno rozdělení souborů pro jednotlivé roky. V letech 2017, 2019 a 2020 je patrný značný výskyt odlehlých hodnot. Průměry se postupně v průběhu let zvyšují. To samé platí i pro mediány, kde se hodnoty taktéž zvyšují. Hodnoty průměrů a mediánů se s malou odchylkou přibližně rovnají. V roce 2018 se průměrná hodnota vyskytuje nad mediánem ve více patrné míře, konkrétněji o 3 %.

4.2 Výpočet pomocí Microsoft Azure Machine Learning Studia

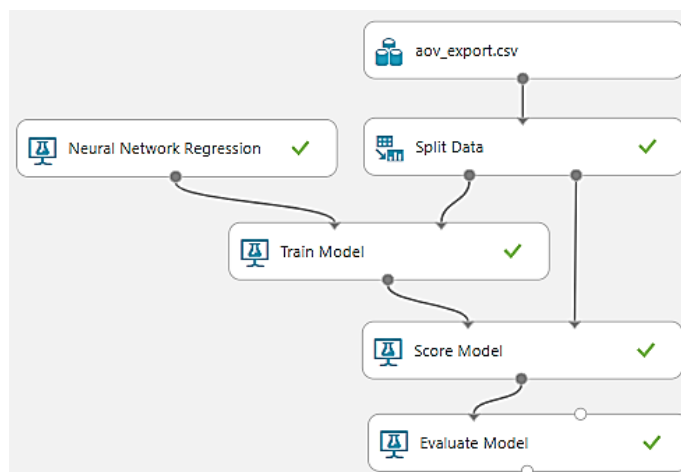
Jak již bylo zmíněno v části metodické, MAMLS umožňuje výpočet regrese pomocí neuronových soustav. Jednou z nezmíněných výhod je tvorba grafické vizualizace prováděných experimentů. Ty jsou rozděleny na trénovací (training experiment, dále jen TE) a prediktivní experiment (predictive experiment, dále jen PE). PE se automaticky vytváří podle nastavení TE, ale je umožněno do něj zasahovat a případně upravovat dle potřeby.

4.2.1 Trénovací experiment

Sestavení trénovacího experimentu je tedy založeno na následujících částech, které se postupně sestaví a mezi sebou propojí. Postup lze krátce shrnout pomocí pěti kroků.

1. Vložení souborů dat ve formátu csv
2. Rozdělení dat na trénovací a testovací (Split Data)
3. Trénování modelu na který je připojena NN regrese. (Train Model)
4. Score Model
5. Vyhodnocení modelu (Evaluate Model)

Obrázek 16: Diagram trénovacího experimentu

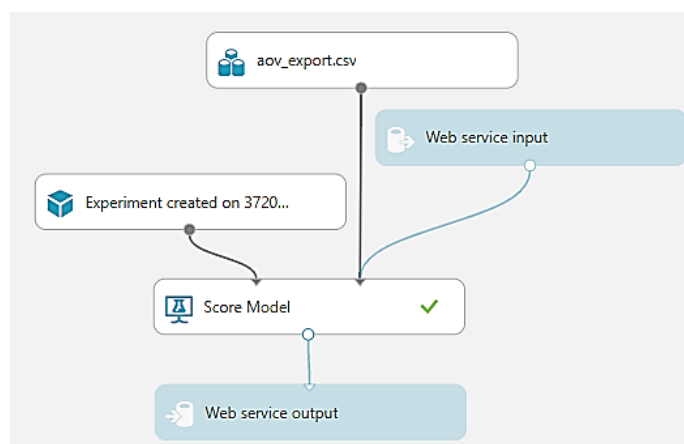


Zdroj: vlastní zpracování

4.2.2 Prediktivní experiment

Ten je vytvářen automaticky a jsou na něj napojeny služby webového rozhraní pro testování modelu (Web service input/output). Dále je díky tomu umožněno pracovat v testovacím režimu.

Obrázek 17: Diagram prediktivního experimentu



Zdroj: vlastní zpracování

Finální výsledky nebyly uspokojivé a například RMSE vycházelo pokaždé nad 10 jednotek (minimální dosažené pak bylo 9.8599 jednotek). Velkou překážkou se stala omezenost možnosti volby nastavení pro tento typ modelu. Ačkoli je možnost práce s programovacím jazykem Python, není toto prostředí zcela uživatelsky přívětivé. Díky tomuto zjištění se změnila strategie postupu a dále se pokračovalo mimo prostředí MAMLS.

4.3 Výpočet modelu LSTM

V této kapitole bude postupováno podle kroků, které jsou zmíněné v kapitole 2.2.3. Prohlášení o nalezení optimálního modelu není u práce s neuronovými sítěmi zcela relevantní. Lze se pouze zamýšlet nad použitím toho nejužitečnějšího, a to ve smyslu, který nejvíce „padne“ v podobnosti k datům, respektive bude nejlépe aproximovat daný soubor dat. Pro výpočet byl vybrán specifický Long-short term memory model. O tom, jak funguje se lze dočíst v kapitole 3.1.9.

4.3.1 Příprava dat

Příprava dat je pro všechny použité konfigurace sítě stejná, a proto není nezbytně nutné tyto postupy komentovat zvlášť.

Jak již bylo zmíněno v kapitole 2.2.1.4, pro práci s daty v prostředí Jupyter Notebook za pomoci programovacího jazyka Python je načtení používaných knihoven nezbytností, ty pak pomůžou jednak se správou či úpravou dat, a samozřejmě i načtení modelů LSTM. Algoritmus pro vložení knihoven je uveden v následující části. Jednotlivé hlavní knihovny jsou popsány v kapitole 2.2.2.

Tabulka 2: Implementace používaných knihoven

In	<pre>import tensorflow as tf import keras import numpy as np import pandas as pd import matplotlib.pyplot as plt from keras.models import Sequential from keras.layers import Dense from keras.layers import LSTM from keras.models import load_model from keras.callbacks import EarlyStopping from keras.callbacks import ModelCheckpoint from sklearn.preprocessing import MinMaxScaler</pre>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Zdroj: vlastní zpracování

Jelikož některé názvy knihoven jsou zdlouhavé, tak se v praxi ustálily používané zkratky, které jsou všeobecně používané. Každopádně není nutnou podmínkou je takto definovat, nebo používat. Příkladem pak může být, pro knihovnu Pandas, zkratka pd, pro Tensorflow tf a podobně. Používání těchto zkratkou mnohdy velice usnadní a urychlí práci s knihovnami.

Na vizualizaci dat bylo pracováno především v softwaru PowerBi kde pak následně byl pomocí MS Excel doupraveny, nicméně pomocí výše zmíněné knihovny Matplotlib lze zobrazovat grafy i přímo v průběhu práce v prostředí Jupyter Notebook. Takto lze poměrně v krátkém čase zkontrolovat, zdali jsou například data nahrány správně. Případné vypsání základních statistických charakteristik datového souboru může být taktéž k užitku při dalším rozhodování, jak s daty pracovat. K tomu, v případě této práce, byla využita především knihovna Pandas.

V Tabulka 3 je uveden zápis pro určení výšky a šířky grafů, aby se zjednodušila práce s nastavováním u jednotlivých grafů v průběhu práce. Používanou zkratkou pro knihovnu Matplotlib bude plt. Grafy jsou nastaveny na šířku 15 a výšku 5 v palcích.

Tabulka 3: Nastavení šířky a výšky grafů

In	<code>plt.rcParams["figure.figsize"] = (15,5)</code>
-----------	------------------------------------------------------

Zdroj: vlastní zpracování

Následující funkce umožní načtení souboru z počítače. Soubor je ve formátu Comma-separated values (dále jen CSV). Oproti jiným formátům se právě tento stává velice důležitý, neboť dokáže pracovat s různými softwary a běžně se nestává, že by nastávala nějaká komplikace kvůli převodu do jiného formátu. Proto se CSV formát stal velice populárním i u datových analytiků.

Tabulka 4: Načtení csv souboru

In	<code>df = pd.read_csv(r'C:\...\main.csv')</code>
-----------	---------------------------------------------------

Zdroj: vlastní zpracování

Nahráný soubor bude nazván df (dataframe). Knihovna Pandas nabízí celou řadu užitečných funkcí pro importování dat. Lze například pracovat i s aktuálními daty od společností jako jsou například Yahoo Finance, nebo Amazon Finance. To je například velkou výhodou při práci s akciovými trhy. Přesný výskyt, respektive cesta k souboru byla skryta. Podobná syntaxe byla využita i při uložení dat zpět do osobního počítače.

Zda import proběhl úspěšně, lze zkontrolovat více způsoby. V prvním případě stačí vpsat název nahraného souboru čili „df“. Výhodou těchto knihoven je, že se vypíše nikoliv celý soubor, ale prvních pět a posledních pět řádků. Existují potom samozřejmě upřesňující funkce, které například vypíší určitý počet řádků od shora (funkce, head()) a jiné.

Tabulka 5: Zobrazení souborů dat „df“

In	<code>df</code>
-----------	-----------------

Out	Date	Bookings	Transactions
0	1/1/2016	38657.855330	965
1	1/2/2016	40111.115390	969
2	1/3/2016	41925.298360	992
3	1/4/2016	35277.811560	914
4	1/5/2016	39946.806760	952
...
1822	12/27/2020	5726.170518	93
1823	12/28/2020	5518.944252	129
1824	12/29/2020	5558.503156	105
1825	12/30/2020	5718.022178	125
1826	12/31/2020	4969.067517	80
1827 rows × 3 columns			

Zdroj: vlastní zpracování

Vypadá to, že se soubor nahrál správně. To ale ještě nemusí znamenat správné naformátování dat a zároveň se v souboru můžou vyskytovat data s chybnou nebo žádnou hodnotou. K zpřesnění údajů o formátování souboru poslouží následující funkce.

Tabulka 6: Základní informace o souboru „df“

In	df.info()
Out	<pre><class 'pandas.core.frame.DataFrame'> RangeIndex: 1827 entries, 0 to 1826 Data columns (total 3 columns): # Column Non-Null Count Dtype --- - 0 Date 1827 non-null object 1 Bookings 1827 non-null float64 2 Transactions 1827 non-null int64 dtypes: float64(1), int64(1), object(1) memory usage: 42.9+ KB</pre>

Zdroj: vlastní zpracování

Z výstupu vyplývá, že jsou načteny tři sloupce datového typu object, float64 a int64. Datový typ object je analogický k datovému typu string. Odlišnost se vyskytuje v případě, pokud se v jednom sloupci vyskytuje více různých typů (například string i integer), potom je celý sloupec automaticky nastaven na právě tento datový typ. Float64 a int64 jsou rozšířené datové typy pro číselné hodnoty s desetinným číslem a bez něj. 64 odkazuje na přidělenou paměť.

Pro zjištění základních statistických charakteristik souboru dat je vhodná následující funkce:

Tabulka 7: Výpis základních charakteristik souboru df

In	df.describe()		
Out		Bookings	Transactions
	count	1827.000000	1827.000000
	mean	12774.535520	326.220580
	std	7556.376824	243.514006
	min	1029.867917	34.000000
	25%	6724.587331	125.000000
	50%	10576.677900	210.000000
	75%	17559.324210	508.000000
	max	41925.298360	1065.000000

Zdroj: vlastní zpracování

Počet hodnot u transakcí je roven počtu hodnot u tržeb. Jsou zde také k vidění průměry, minimum a maximum, medián, horní či dolní kvartily. Standardní odchylka je dosti vysoká v obou případech. Pro počet transakcí na den až 244 a u tržeb na den se hodnota blíží až 7600 USD. Podrobnější analýza byla uvedena v předchozí kapitole 4.1.2. Vzhledem ke shodám v podobnosti charakteristik souboru se softwaru Power BI, lze usuzovat o správnosti načtení dat.

4.3.2 Příprava časové řady a výpočet AOV

Na vstupu do modelu LSTM jsou pro případ této práce požadovány data jednodimenzionální. Tak byla i stanovena strategie práce s daty, kdy jako hlavní vstupní proměnné do tohoto modelu je AOV, jež je dáno podílem tržeb a transakcí. Do tabulky „df“ je tedy vložen další sloupec, pojmenovaný jako „AOV“ a do něj jsou vloženy hodnoty podílu. Syntaxe pro tento postup je následující.

Tabulka 8: Výpočet AOV

In	df["AOV"] = df["Bookings"]/df["Transactions"] df

Out	Date	Bookings	Transactions	AOV
0	1/1/2016	38657.855330	965	40.059954
1	1/2/2016	40111.115390	969	41.394340
2	1/3/2016	41925.298360	992	42.263406
3	1/4/2016	35277.811560	914	38.597168
4	1/5/2016	39946.806760	952	41.960931
...
1822	12/27/2020	5726.170518	93	61.571726
1823	12/28/2020	5518.944252	129	42.782514
1824	12/29/2020	5558.503156	105	52.938125
1825	12/30/2020	5718.022178	125	45.744177
1826	12/31/2020	4969.067517	80	62.113344
1827 rows × 4 columns				

Zdroje: vlastní zpracování

Jak již bylo zmíněno, je nutné docílit, aby data byla jednodimenzionální čili aby obsahovaly pouze samotné AOV. Toho lze docílit jednak převedením časových hodnot na indexy a odstraněním sloupců s transakcemi a tržbami. To vše lze udělat následujícím jednoduchým zápisem.

Tabulka 9: Změna formátu datumu, převedení na index a zobrazení informací o souboru

In	<pre>df['Date'] = pd.to_datetime(df['Date'],format='%m/%d/%Y').dt.strftime('%Y-%m-%d') df = df.set_index(['Date']) df2 = df.drop(columns=['Transactions', 'Bookings']) df2.info()</pre>
Out	<pre><class 'pandas.core.frame.DataFrame'> Index: 1827 entries, 2016-01-01 to 2020-12-31 Data columns (total 1 columns): # Column Non-Null Count Dtype --- - 0 AOV 1827 non-null float64 dtypes: float64(1) memory usage: 28.5+ KB</pre>

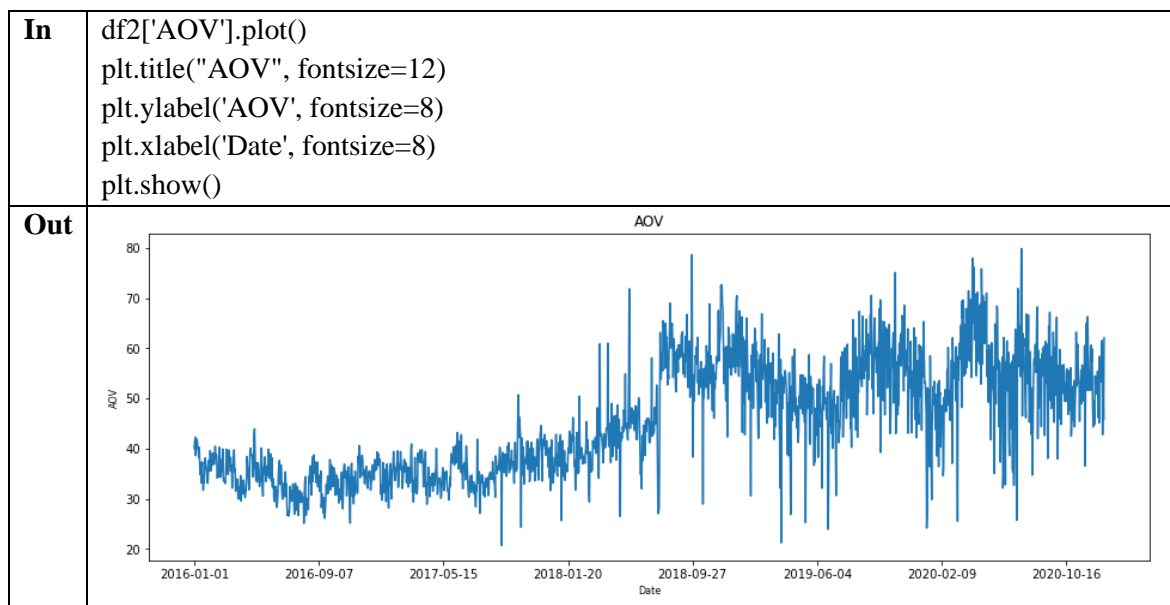
Zdroje: vlastní zpracování

Podle výstupu je patrná změna indexů. Ta je teď nastavena na časovou řadu od počátku roku 2016 až do konce roku 2020. Počet sloupců je pouze jeden se 1827 nenulovými hodnotami. Datový typ je nastaven tak, aby bral v potaz i desetinné číslo.

4.3.2.1 Popis souboru df2 a jeho grafické vyjádření

Grafické zobrazení časové řady pro AOV jsou znázorněny, pomocí funkcí načtených z knihovny Matplotlib následně.

Tabulka 10: Grafické zobrazení AOV



Zdroj: vlastní zpracování

Díky nastavení indexů a nastavení formátování datumu se automaticky nastavila horizontální osa na datum. Vypadá to, že se tento graf s výstupem z programu PowerBI shoduje. Nicméně je důležité ještě zkontrolovat základní statistické údaje. Znovu tedy bude aplikována funkce pro popis základních statistických charakteristik, tentokrát pro soubor df2.

Tabulka 11: Charakteristiky souboru dat df2

In	<pre>df2['AOV'].describe()</pre>
Out	<pre>count 1827.000000 mean 45.100204 std 11.144883 min 20.703274 25% 35.485952 50% 42.950448 75% 54.397624 max 79.910060 Name: AOV, dtype: float64</pre>

Zdroj: vlastní zpracování

Z popisné statistiky vyplývá, že se maximální hodnota blíží 80 USD a minimální je přibližně 21 USD. Průměrná hodnota je rovna přibližně 45 USD. Rozdíl mezi aritmetickým průměrem a mediánem je nepatrný, a proto se dá předpokládat, že se zde nevyskytují odlehlá čísla, která by průměr ve výsledku značně zkreslovala. Průměrná cena celého souboru je přibližně 45 USD na den na jednu transakci. Směrodatná odchylka je přibližně 11 USD. To vyplývá i z kapitoly 4.1.2.3.

4.3.3 Trénování modelu

Dalším krokem je samotné trénování modelu. Na to je potřeba rozdělit základní soubor s hodnotami AOV na trénovací a testovací. Obecně platí, že se základní soubor rozděluje ze 70 % na trénovací soubor a ze zbylých 30 % na testovací soubor.

Tabulka 12: Rozdělení souboru df2 na trénovací a testovací

In	<pre>trdf2 = df2[:1274] print('Trenovaci:', trdf2.shape) tedf2 = df2[1274:] print('Testovaci: ',tedf2.shape)</pre>
Out	<pre>Trenovaci: (1274, 1) Testovaci: (553, 1)</pre>

Zdroj: vlastní zpracování

Z výstupu je patrné to, že základní soubor dat df2 byl rozdělen na 1274 (70 %) trénovacího souboru a 553 (30 %) testovací souboru.

Pro práci s modely LSTM musí být každý údaj převeden na číslo patřícího do intervalu mezi 0 a 1. K tomu se používá funkce MinMaxScaler, která dokáže tato data v daném poměru transformovat. Trénovací soubor dat může být například normalizovaný pomocí tohoto zápisu.

Tabulka 13: Použití funkce MinMaxScaler()

In	<pre>scaler = MinMaxScaler() tr = scaler.fit_transform(trdf2.values.reshape(-1, 1)) te = scaler.transform(tedf2.values.reshape(-1, 1)) print("MIN Trenovaci:", np.min(tr)) print("MAX Trenovaci:", np.max(tr))</pre>
Out	<pre>MIN Trenovaci: 0.0 MAX Trenovaci: 1.0</pre>

Zdroj: vlastní zpracování

Je důležité nejprve model přizpůsobit testovacím datům a poté transformovat testovací data za použití tohoto stejného modelu. To je učiněno pomocí funkcí `fit_transform` a `transform`. V jiném případě může dojít ke špatným výsledkům.

Poslední příprava, která zbývá, před samotným spuštěním trénování je převedení souboru dat na sekvenční data. Pro lepší pochopení je uveden zjednodušený příklad.

Práce se sekvenčními daty je velice užitečná zvláště pak s daty, kde je určitá závislost na předchozích časových krocích.

Tabulka 14: Sekvenční postup výpočtu

Krok	Vstup – sekvence	Výstup – jedna hodnota
1	4, 5, 6	5
2	5, 6, 5	8
3	6, 5, 8	6

Zdroj: vlastní zpracování

Na základě sekvence dat minulých vstupů do modelu je vygenerována jedna hodnota na výstupu. Pokud tedy na vstupu pro první krok je sekvence 4, 5, 6, tak výstupem je predikovaná hodnota 5. V druhém kroku se Vstupní sekvence posune o jedno místo doprava a přiřadí novému poli hodnotu predikovanou v kroku 1. Potom je tedy v kroku 2 na vstupu sekvence 5, 6, 5 a je predikován výstup 8. A takto je postupováno dál. V kroce tři je potom vstupní sekvence 6, 5, 8 a její predikovaná hodnota 6.

Stejný postup je zapsán v následujícím kódu akorát je sekvence nastavena na 7 předcházejících dnů.

Tabulka 15: Algoritmus pro vytvoření sekvencí

In	<pre>def split(sequence, steps): x, y = [], [] for i in range(len(sequence)): endix = i + steps if endix > len(sequence)-1: break seqx, seqy = sequence[i:endix], sequence[endix] x.append(seqx) y.append(seqy) return np.array(x), np.array(y) tr steps = 7 xtr, ytr = split(tr, steps) xtr.T</pre>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Out	<pre>array([[0.33390091, 0.35691894, 0.37191024, ..., 0.6509795 , 0.48886844, 0.50196647], [0.35691894, 0.37191024, 0.30866799, ..., 0.48886844, 0.50196647, 0.50763102], [0.37191024, 0.30866799, 0.36669259, ..., 0.50196647, 0.50763102, 0.4655652], ..., [0.36669259, 0.36491905, 0.36107531, ..., 0.4655652 , 0.43322803, 0.4378358], [0.36491905, 0.36107531, 0.33293093, ..., 0.43322803, 0.4378358 , 0.52821004], [0.36107531, 0.33293093, 0.33006986, ..., 0.4378358 , 0.52821004, 0.05545743]])</pre>
------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Zdroj: (Sadawi, 2020)

Z výstupu této části syntaxe je mnohem více zřejmé, že se všechna data nacházejí v intervalu od 0 do 1. Knihovny Pandas pracují s array (pole). To umožní do nově vzniklých proměnných uložit sekvenční data. Jak tomu je i na příkladu výstupu transponované proměnné xtr. Sekvence jsou nastaveny na 7 kroků, respektive dní. Proměnné X a Y jsou vstupy a výstupy výše zmíněných sekvencí.

Obdobně jako bylo pracováno se souborem trénovacích dat se tento algoritmus aplikuje i pro data testovací.

Tabulka 16: Zobrazení struktury trénovacího a testovacího souboru

In	<pre>xte, yte = split(te, steps) print(xtr.shape) print(xte.shape)</pre>
Out	<pre>(1267, 7, 1) (546, 7, 1)</pre>

Zdroj: vlastní zpracování

Nyní jsou data třírozměrná a podle výstupu z funkce split, je patrné rozdělení. V proměnných xte a yte se nyní nachází 1267, respektive 546 sekvencí po 7 hodnotách v každé z nich. Respektive se dá mluvit o vektorech.

4.3.4 Výpočet modelu LSTM

Doposud docházelo pouze k přípravě dat, která jsou vstupem do výpočtu rekurentního modelu LSTM.

4.3.4.1 Příklad výpočtu vybrané modifikace modelu LSTM

Příklad výpočtu pro konfiguraci sítě 16-32-64-1 může vypadat například takto:

Tabulka 17: Algoritmus pro výpočet modelu LSTM

In	<pre> model = Sequential() model.add(LSTM(16, activation='relu', input_shape=(steps, n), return_sequences=True)) model.add(LSTM(32, activation='relu', return_sequences=True)) model.add(LSTM(64, activation='relu')) model.add(Dense(1)) model.compile(optimizer='adam', loss='mse') es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=30) mc = ModelCheckpoint('lstm_model_finale.h5', monitor='val_loss', mode='min', verbose=1, save_best_only=True) mf = model.fit(xtr, ytr, validation_split=0.2, epochs=200, batch_size=64, verbose=1, callbacks=[es, mc]) </pre>
Out	<pre> Epoch 1/200 14/16 [=====>...] - ETA: 0s - loss: 0.0866 Epoch 00001: val_loss improved from inf to 0.17540, saving model to lstm_model_finale.h5 16/16 [=====] - 1s 92ms/step - loss: 0.0813 - val_loss: 0.1754 Epoch 2/200 15/16 [=====>..] - ETA: 0s - loss: 0.0226 Epoch 00002: val_loss improved from 0.17540 to 0.03516, saving model to lstm_model_finale.h5 16/16 [=====] - 0s 30ms/step - loss: 0.0226 - val_loss: 0.0352 Epoch 3/200 15/16 [=====>..] - ETA: 0s - loss: 0.0153 Epoch 00003: val_loss did not improve from 0.03516 16/16 [=====] - 0s 25ms/step - loss: 0.0155 - val_loss: 0.0624 ... Epoch 133/200 15/16 [=====>..] - ETA: 0s - loss: 0.0044 Epoch 00133: val_loss did not improve from 0.01556 16/16 [=====] - 0s 27ms/step - loss: 0.0043 - val_loss: 0.0170 Epoch 134/200 16/16 [=====] - ETA: 0s - loss: 0.0042 Epoch 00134: val_loss did not improve from 0.01556 16/16 [=====] - 0s 25ms/step - loss: 0.0042 - val_loss: 0.0168 Epoch 135/200 15/16 [=====>..] - ETA: 0s - loss: 0.0042 Epoch 00135: val_loss did not improve from 0.01556 16/16 [=====] - 0s 26ms/step - loss: 0.0043 - val_loss: 0.0169 Epoch 00135: early stopping </pre>

Zdroj: vlastní zpracování

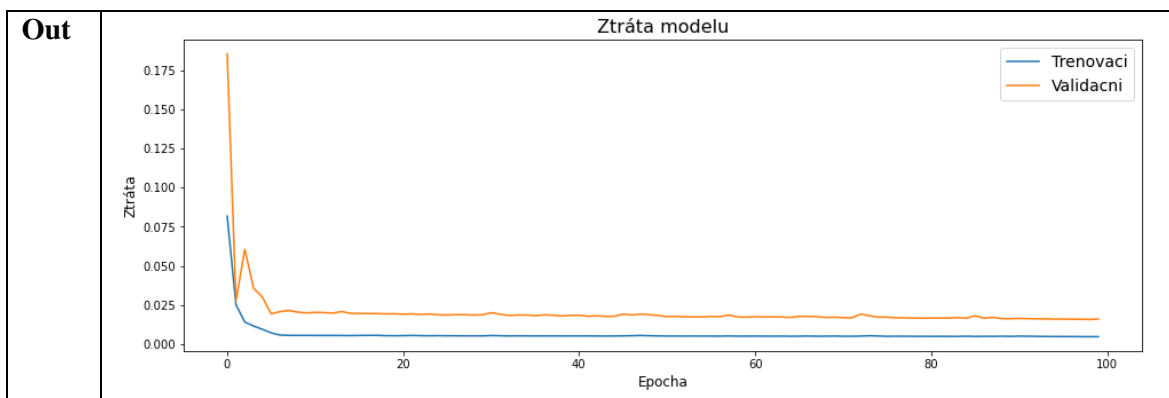
Jako první je řečeno modelu, že je pracováno se sekvenčními daty (funkce `Sequence()`). Ve druhém až pátém řádku jsou nadefinovány požadavky pro jednotlivé vrstvy. Ve vstupní vrstvě je 16 neuronů. Jako další je nadefinována aktivační funkce čili ReLU. Její popis je zmíněn v teoretické části. Stejně jako jsou na vstupu do vstupní neuronové sítě sekvenční data, tak i výstupem z této sítě musí být sekvence dat (proto je `return_sequence` nastaven na hodnotu `true`). Obdobou jsou skryté sítě, kdy v jednom případě bylo zvoleno 32 neuronů a ve druhém 64 neuronů. Výstupní síť má pouze jeden neuron, protože pracujeme s jednodimenzionálními daty. Důležitou funkcí je funkce `model.fit`, ta do velké míry

ovlivňuje průběh výpočtů modelu. Rozdělení trénovacích dat na validační je 20 %. Počet epoch byl nastaven na 200. Počet výchozích trénovacích hodnot (32) použitých v jedné interaci byl zvýšen na 64. Verbose byl nastaven na 1 díky tomu je viditelný průběh výpočtů pro jednotlivé epochy. Tak jak je tomu ve výstupu z modelu. Aby se zamezilo přeučení, byl vložen algoritmus, který tomu zamezí (funkce EarlyStopping()). Tato funkce je nastavena tak, aby sledovala ztráty na validačním souboru (val_loss). Dále je specifikován argument pro minimalizaci středních kvadratických odchylek validačního souboru (mode je nastaven na min). Přidáním argumentu „patience“ (trpělivost) je docíleno zpoždění funkce předčasného ukončení, neboť mnohdy dobrým výsledkům předchází i hodnoty špatné v průběhu interací. To ale nutně neznamená, že se model nemusí zlepšovat. Například v tomto vybraném příkladě by jinak výpočet skončil už u třetí epochy, neboť je zde zmíněno „val_loss did not improve“ (nedošlo ke zlepšení ve ztrátách na validačním souboru). Funkce předčasného ukončení nakonec proces výpočtů ukončila, ale až po 135 epoše. Poslední nevysvětlenou zůstává funkce „ModelCheckpoint()“. Ta zajistí, aby byl uložen model, který v průběhu trénování vykazuje nejlepší výsledky. Uloží ho do speciálního formátu h5, respektive v tomto konkrétním případě jako „lstm_model_finale.h5“.

Proces trénování modelu, je možné díky knihovně Keras, zobrazit graficky. Potom tedy na horizontální ose jsou vyjádřeny epochy a na vertikální ose ztráta. Porovnávají se zde dva soubory, a to trénovací a validační. Algoritmus pro zobrazení může vypadat například takto.

Tabulka 18: Algoritmus pro zobrazení ztráty modelu

In	<pre>plt.plot(mf.history['loss']) plt.plot(mf.history['val_loss']) plt.title('Ztráta modelu') plt.ylabel('Ztráta') plt.xlabel('Epocha') plt.legend(['Trenovaci', 'Validacni'], loc='upper left') plt.show()</pre>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Zdroj: vlastní zpracování

Výstupem této syntaxe kódu je grafické vyjádření průběhu učení. V dalším porovnávání o kvalitě naučeného modelu je na místě porovnávat dvě křivky reprezentující trénování a validaci. Přesné vysvětlení, jak mezi jednotlivými křivkami rozpoznat predikční schopnost modelu je popsáno v kapitole 3.1.10. V grafu na horizontální ose se nachází počet epoch. V případě nastavení rekurentní sítě LSTM je stanoven na maximální hodnotu 200 epoch. Na vertikální ose se pak vyjadřuje ztráta. V tomto případě model nemá velmi dobré predikční schopnosti, protože křivka vyjadřující trénovací data nemá stejný průběh a zároveň se nepřekrývají. Model nemá ani velmi špatné predikční schopnosti, neboť by vzdálenost mezi křivkou trénovací a testovací musela být nesmyslně veliká. Pokud model není dokonalý, ale zároveň není špatná nastává případ, kdy se o schopnosti modelu predikovat uspokojivě.

Tabulka 19: Načtení modelu a predikce hodnot

In	<pre> model = load_model('lstm_model_finale.h5') pred = [] for i in range(xte.shape[0]): xin = xte[i] xin = xin.reshape((1, steps, n)) yhat = model.predict(xin, verbose=0) pred.append(yhat[0]) print(yhat[0], yte[i]) </pre>
Out	<pre> [0.46660686] [0.33402128] [0.43585035] [0.49194393] [0.44647834] [0.51620304] [0.4607641] [0.53209004] [0.47610965] [0.48299519] ... </pre>

	[0.5783545] [0.70497696] [0.6143232] [0.38086481] [0.55105114] [0.55604816] [0.5363373] [0.43195323] [0.5056591] [0.71431982]
--	-------------------------------------------------------------------------------------------------------------------------------------------

Zdroj: vlastní zpracování

V kapitole 4.3.2, byly všechny hodnoty převedeny do intervalu od 0 do 1. Proto i výše uvedené hodnoty se budou nacházet v tomto rozmezí. Pro získání skutečných hodnot, jsou převedeny pomocí inverzní funkce „scaler“ zpět do původní podoby, tak aby vyjadřovali skutečné hodnoty AOV v amerických dolarech.

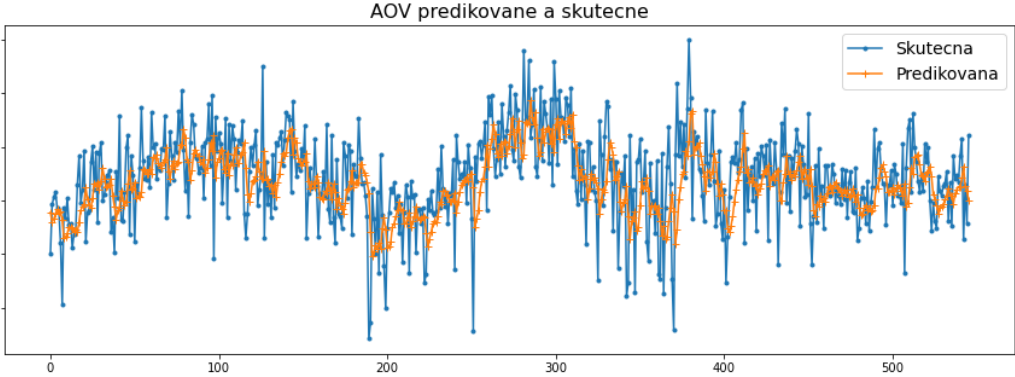
Tabulka 20: Inverzní transformace hodnot

In	<pre> pred = scaler.inverse_transform(np.array(pred)) actual = scaler.inverse_transform(yte) df2pred = pd.DataFrame(pred) df2pred.columns = ['Predicted Value'] df2actual = pd.DataFrame(actual) df2actual.columns = ['Actual Value'] df2exp = pd.concat([df2pred, df2actual], axis=1) df2exp </pre>																																				
Out	<table border="1"> <thead> <tr> <th></th> <th>Predicted Value</th> <th>Actual Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>47.753094</td> <td>40.066932</td> </tr> <tr> <td>1</td> <td>45.970100</td> <td>49.221918</td> </tr> <tr> <td>2</td> <td>46.586216</td> <td>50.628251</td> </tr> <tr> <td>3</td> <td>47.414379</td> <td>51.549241</td> </tr> <tr> <td>4</td> <td>48.303986</td> <td>48.703148</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>541</td> <td>54.231251</td> <td>61.571726</td> </tr> <tr> <td>542</td> <td>56.316406</td> <td>42.782514</td> </tr> <tr> <td>543</td> <td>52.648441</td> <td>52.938125</td> </tr> <tr> <td>544</td> <td>51.795460</td> <td>45.744177</td> </tr> <tr> <td>545</td> <td>50.017002</td> <td>62.113344</td> </tr> </tbody> </table> <p>546 rows × 2 columns</p>		Predicted Value	Actual Value	0	47.753094	40.066932	1	45.970100	49.221918	2	46.586216	50.628251	3	47.414379	51.549241	4	48.303986	48.703148	541	54.231251	61.571726	542	56.316406	42.782514	543	52.648441	52.938125	544	51.795460	45.744177	545	50.017002	62.113344
	Predicted Value	Actual Value																																			
0	47.753094	40.066932																																			
1	45.970100	49.221918																																			
2	46.586216	50.628251																																			
3	47.414379	51.549241																																			
4	48.303986	48.703148																																			
...																																			
541	54.231251	61.571726																																			
542	56.316406	42.782514																																			
543	52.648441	52.938125																																			
544	51.795460	45.744177																																			
545	50.017002	62.113344																																			

Zdroj: vlastní zpracování

Mimo převedení hodnot do původního vyjádření v USD, je zde aplikováno i vytvoření tabulky se dvěma sloupci, jejichž atributy jsou nazvány jako „Predikovaná hodnota“ pro hodnoty predikované a „Aktuální hodnota“ pro hodnoty původní. Tyto výstupní hodnoty potom v konečném výsledku graficky vypadají následovně.

Tabulka 21: Zobrazení grafu predikovaných a skutečných hodnot

In	<pre>plt.title('AOV predikovane a skutecne', fontsize=16) plt.plot(actual, marker='.', label='Skutecna') plt.plot(pred, marker='+', label='Predikovana') plt.legend(fontsize=14)</pre>
Out	 <p>The graph displays two data series: 'Skutecna' (Actual) represented by a blue line with circular markers, and 'Predikovana' (Predicted) represented by an orange line with plus-sign markers. The x-axis represents days from 0 to 545, and the y-axis represents AOV values from 30 to 80. The predicted values track the actual values very closely throughout the period.</p>

Zdroj: vlastní zpracování

Na horizontální ose jsou dny, počítané od 0 až 545. a na vertikální ose je vyneseno AOV v amerických dolarech. Je pozoruhodné, že ačkoli model „viděl“ při výpočtech pouze trénovací data, tak se v tomto případě predikovaná hodnota opravdu snaží skutečná data celkem dobře aproximovat.

4.4 Simulace

Výpočet modelu je opravdu velice subjektivně postaven. A záleží na autorovi daných propočtu, jakým nastavením sítě dá přednos. Vyzkoušení a nastavení všech parametru je nad rámec této práce, a proto se bude upravovat pouze část modelu, na kterém se budou dále vytvářet simulace. Především je pracováno s konfigurací sítě, kde je pozměněn počet neuronů ve vstupní a skrytých vrstvách sítě. Následně je také přidána a ubrána jedna skrytá síť. Beze změny zůstanou například aktivační funkce, počet neuronů ve výstupní vrstvě, rozdělení vstupních dat, počet epoch, rozdělení validačního souboru a nastavení předčasného zastavení.

4.4.1 Změna počtu neuronů ve skrytých vrstvách

Změna je prováděna na skrytých vrstvách. Byla ponechána stejná kalkulace jako v ukázkovém vybraném výpočtu. Sítě tedy obsahují vždy dvojnásobek předchozí vrstvy. To je ostatně i doporučováno odborníky z praxe. Zápis je následující 8-16-32-1. První a poslední se týká počtu neuronů vstupní a výstupní vrstvy sítě. Dvě prostřední jsou skrytými vrstvami.

	Konfigurace sítě I	Konfigurace sítě II
1	4-8-16-1	4-16-8-1
2	8-16-32-1	8-32-16-1
3	16-32-64-1	16-64-32-1
4	32-64-128-1	32-128-64-1

4.4.1.1 První simulace pro konfiguraci sítě 4-8-16-1 a 4-16-8-1

Pro první simulaci byla vybrána struktura sítě 4-8-16-1 a 4-16-8-1. Grafické znázornění vztahů trénovacího a validačního souboru vypadá následovně.

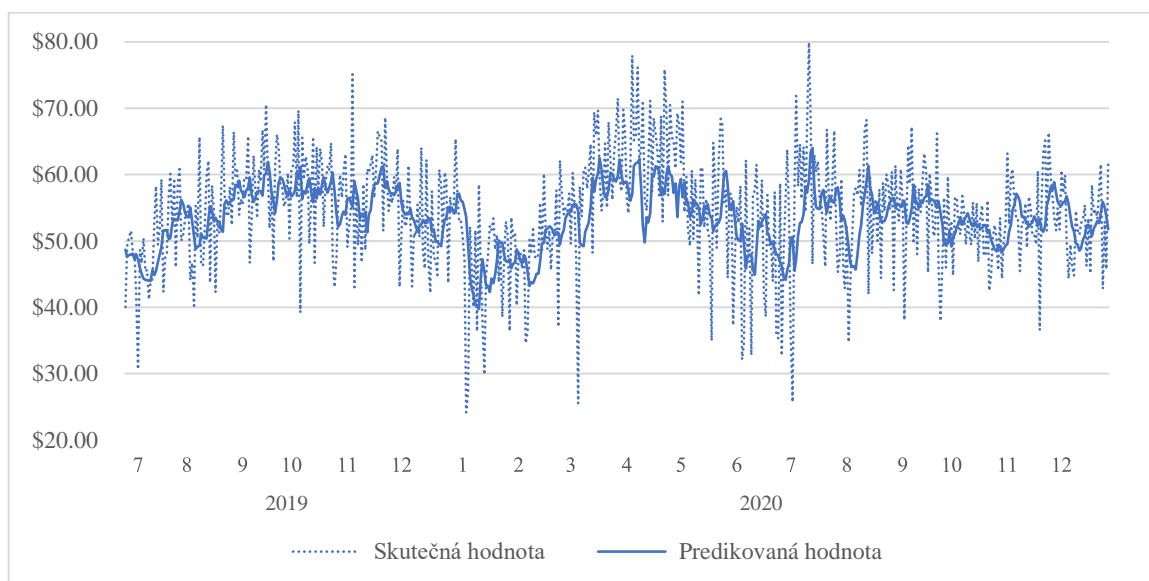
Graf 10: Znázornění ztráty modelu v průběhu epoch pro konfiguraci sítě 4-8-16-1



Zdroj: vlastní zpracování

Z grafu 10 je patrný pokles v průběhu prvních epoch. Následně se validační soubor blíží trénovacímu a při 31 epoše začíná stagnovat. Dále se validační data drží nad trénovacími po celou dobu. Ten dále stagnuje do epochy poslední. Ta je nastavena na 200. Při výpočtu LSTM nebylo využito funkce předčasného ukončení. Predikční schopnost modelu se dá ve výsledku předpokládat jako uspokojivá. Porovnání predikované a skutečné hodnoty je znázorněno níže.

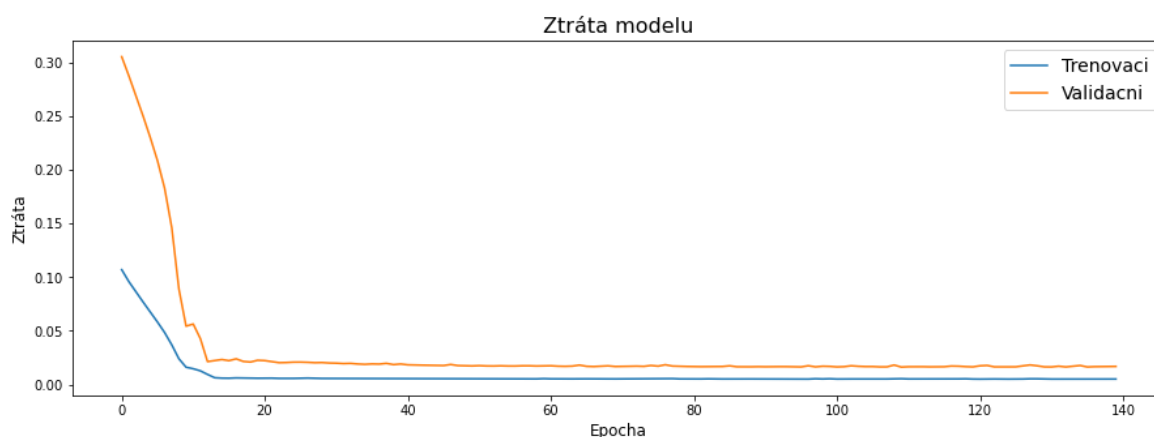
Graf 11: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 4-8-16-1



Zdroj: vlastní zpracování

Rozdíl sumy kvadrátu odchylek je 7,9469 USD. Minimální predikovaná hodnota je přibližně 39,77 USD a maximální je okolo 64,01 USD. Průměr souboru je 53,48 USD. S podobnou konfigurací sítě s prohozeným počtem neuronů skrytých vrstev a to 4-16-8-1 je grafické znázornění ztráty modelu docela jiné.

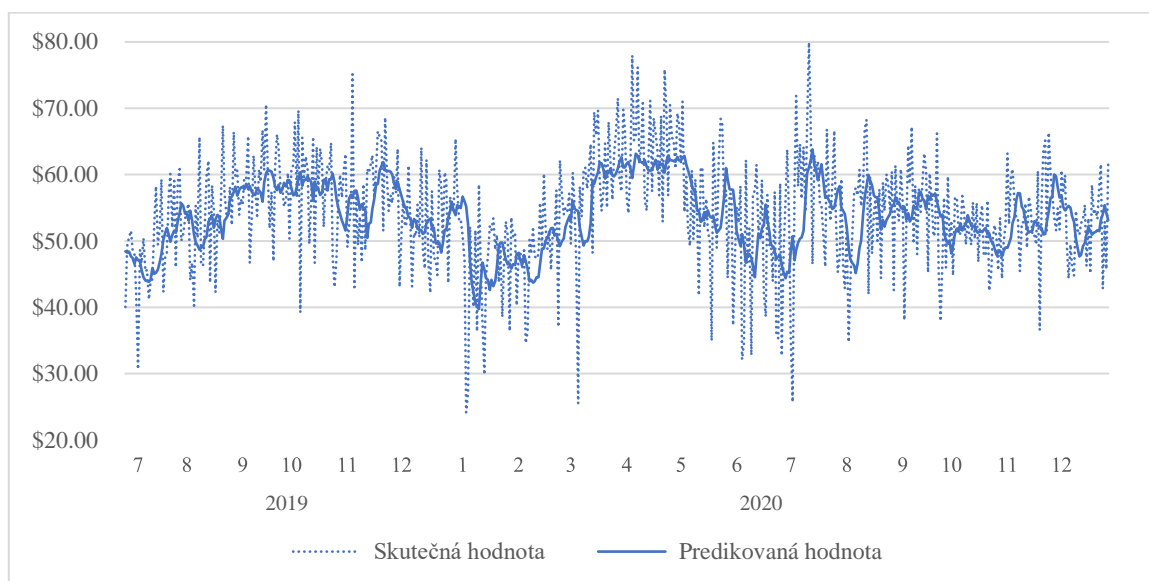
Graf 12: Znázornění ztráty modelu v průběhu epoch pro konfiguraci sítě 4-16-8-1



Zdroj: vlastní zpracování

Výpočet skočil po 140 epochách na základě funkce předčasného ukončení. U čtrnácté epochy hodnoty začínají stagnovat, a to okolo hodnoty 0,0055. Ke konci se hodnoty dostanou až na 0,0050. I přes nejistý začátek v prvních epochách se dá v celku zhodnotit model jako uspokojivý.

Graf 13: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 4-16-8-1



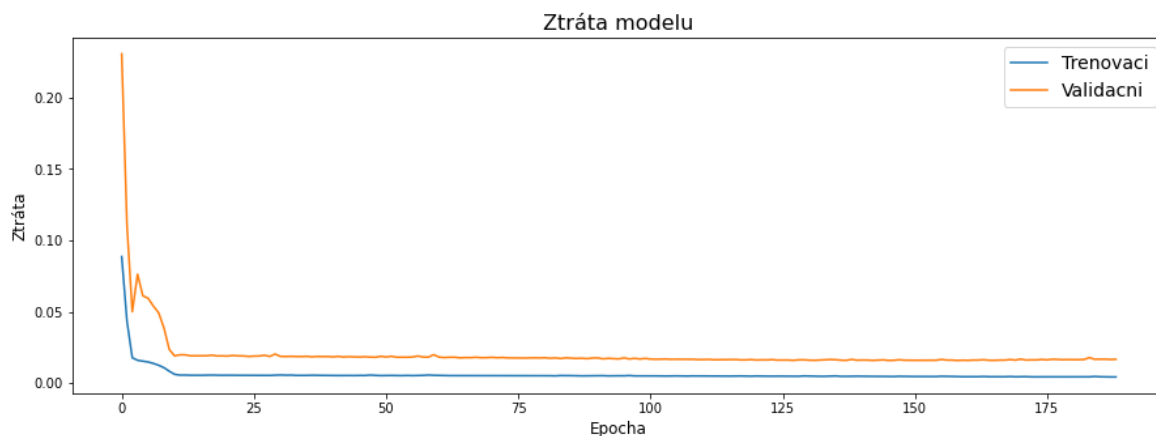
Zdroj: vlastní zpracování

V porovnání s předešlým modelem je RMSE o něco nižší a to 7.8376 USD. To vyplývá i z grafů predikovaných a skutečných hodnot, kde se ve druhém případě dochází k lepší aproximaci skutečných hodnot. Predikované hodnoty se nachází v intervalu od 39,75 USD do 63,80 USD. Průměr je přibližně 53,72 USD.

4.4.1.2 Druhá simulace pro konfiguraci sítě 8-16-32-1 a 8-32-16-1

V druhé simulaci byl navýšen počet neuronů v sítí o polovinu nižší, a to u každé vrstvy z první simulace, kromě výstupní. Konfigurace sítě je potom 8-16-32-1 a 8-32-16-1.

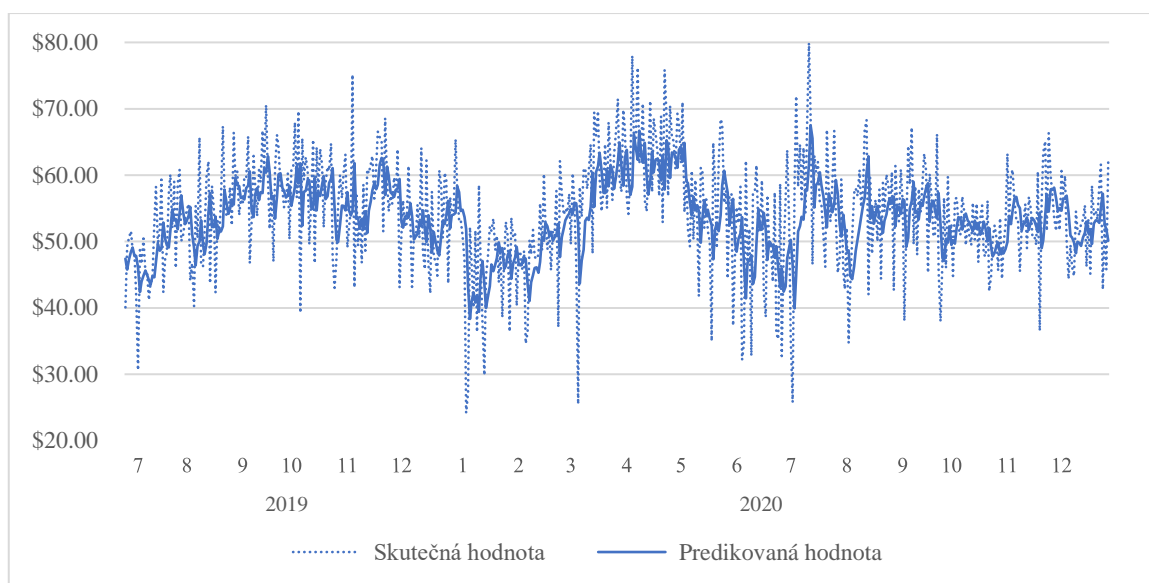
Graf 14: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 8-16-32-1



Zdroj: vlastní zpracování

Vývoj ztráty modelu v průběhu epoch pro model 8-16-32-1 je znázorněn na grafu 14. I v tomto případě je možné brát predikční schopnost grafu jako uspokojivou. O tom vypovídá vzdálenost trénovacího a validačního souboru. Ta je po většinu epoch blízko u sebe. Mezi trénovacím a validačním souboru došlo v průběhu trénování k zvýšení validačního souboru v epochách 4 až 7. Hodnota se poté usazuje a stagnuje okolo 0,0191. U konce nabývá hodnot v průměru 0,0161. U této simulace došlo k aktivaci funkce předčasného ukončení, protože výpočet se zastavil u epochy 189 z 200 možných.

Graf 15: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 8-16-32-1

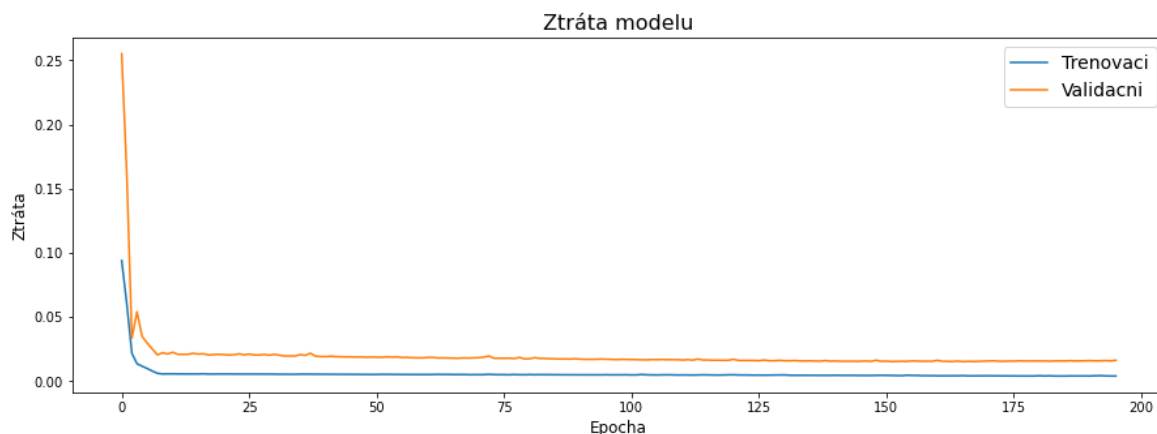


Zdroj: vlastní zpracování

Dle porovnání predikovaných a skutečných hodnot lze usoudit, že byl model dobře natrénován a dokáže tak uspokojivě aproximovat skutečné hodnoty to je vidět z Graf 15. Podle softwaru PowerBI se hodnota RMSE rovná 7,9801 USD. Ta je o něco vyšší než u obou případů konfigurace sítě v první simulaci.

Konfigurací sítě 8-32-16-1 je dosaženo následujícího průběhu trénování modelu.

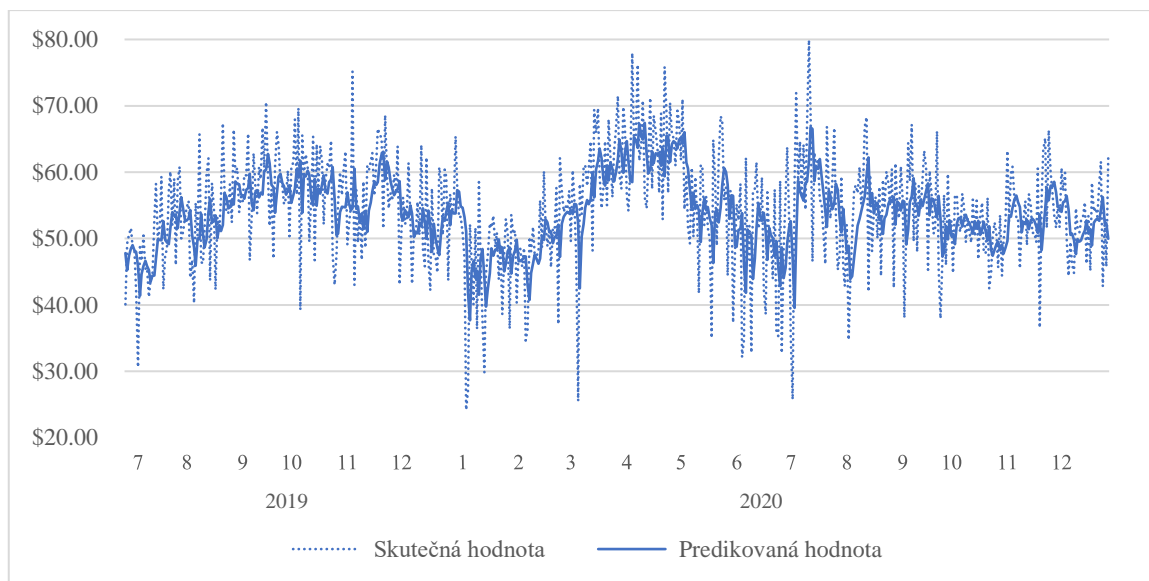
Graf 16 : Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 8-32-16-1



Zdroj: vlastní zpracování

Z grafu 16 je patná uspokojivá predikční schopnost modelu. V průběhu trénování opět dochází k předčasnému ukončení výpočtů, a to v epoše 196. Validační křivka v průběhu trénování sítě vykazuje lokální anomálii opět v epoše čtvrté.

Graf 17 : Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 8-32-16-1



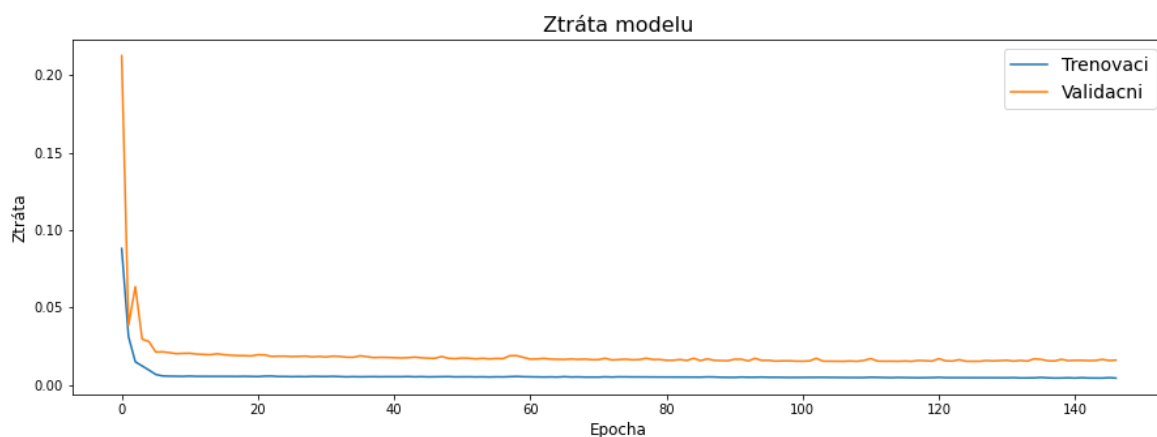
Zdroj: vlastní zpracování

Hodnoty AOV predikovaného souboru dat se pohybují v intervalu mezi dolní hranicí 37,67 USD a horní hranicí 67,43 USD. Průměrná hodnota je, oproti skutečnému souboru dat, nižší, a to 53,76 USD. Zmenšila se i směrodatná odchylka přibližně na hodnotu 5,20 USD. RMSE je v případě této konfigurace 7,8278 USD.

4.4.1.3 Třetí simulace pro konfiguraci sítě 16-32-64-1 a 16-64-32-1

Třetí simulace byla znova navýšena o polovinu neuronů v každé vrstvě, kromě výstupní vrstvy. Simulovány pak byly konfigurace sítě 16-32-64-1 a 16-64-32-1. Průběh trénovací a validační křivky pro konfiguraci první je uveden v grafu 18.

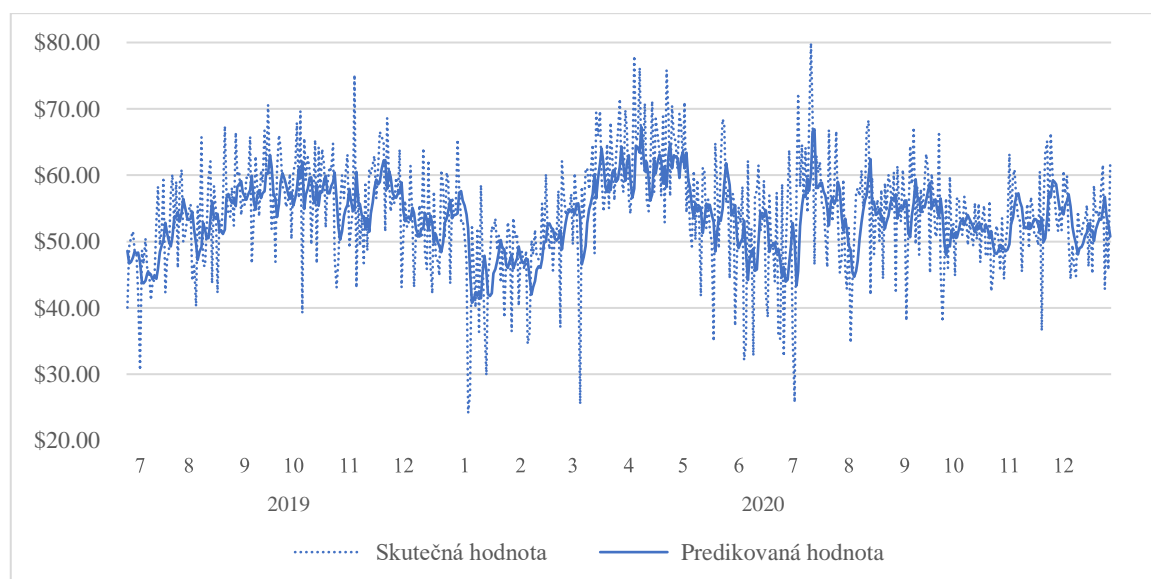
Graf 18: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 16-32-64-1



Zdroj: vlastní zpracování

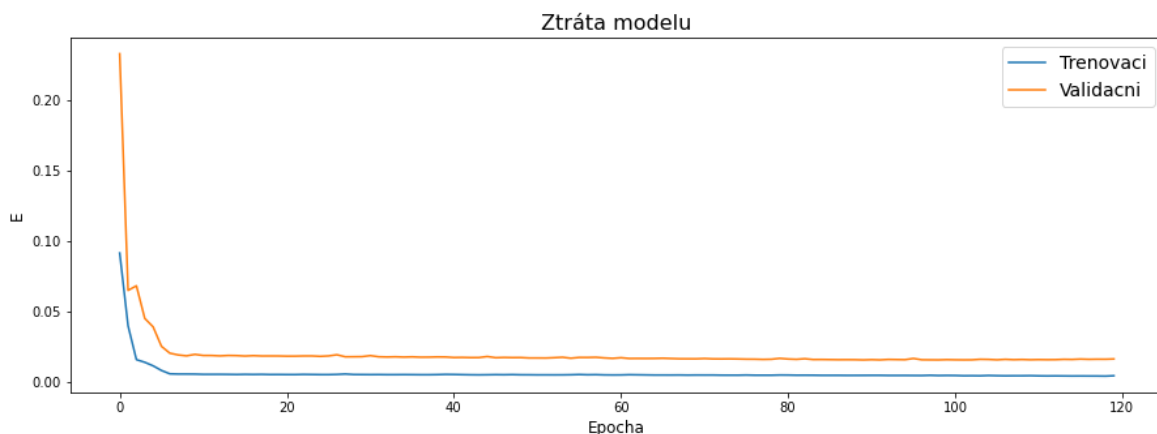
Obě tyto křivky (trénovací a validační) klesají. V epoše 117 pak dochází k aktivaci algoritmu pro předčasné ukončení a po dobu dalších 30 stanovených interací je výpočet zastaven a to ve 147 epoše. Hodnota validační křivky se u konce pohybovala okolo 0,0160. Predikční schopnost lze shrnout jako uspokojivou. RMSE je přibližně 7,8909 USD

Graf 19: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 16-32-64-1



Zdroj: vlastní zpracován

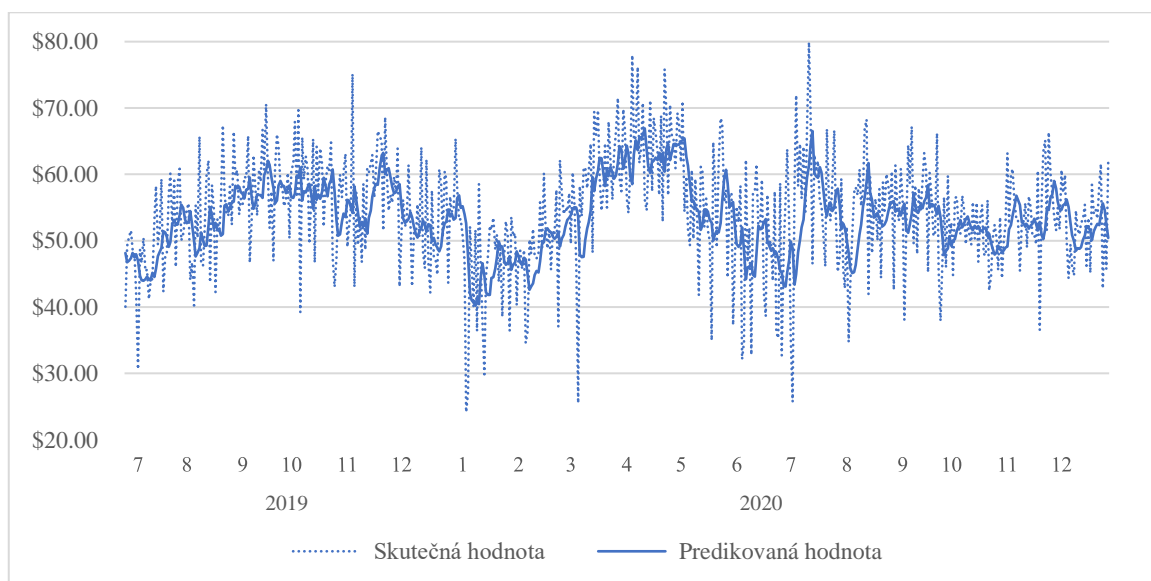
Graf 20: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 16-64-32-1



Zdroj: vlastní zpracování

Validační a trénovací křivka klesá v průběhu interací. Nejdřív strměji do epochy 8 a pak už mírně, a to až do epochy 120, kde byl výpočet předčasně ukončen. V posledních epochách se validační ztráta pohybuje okolo 0,0162. Graf průběhu učení vypovídá o schopnosti modelu uspokojivé predikční schopnosti.

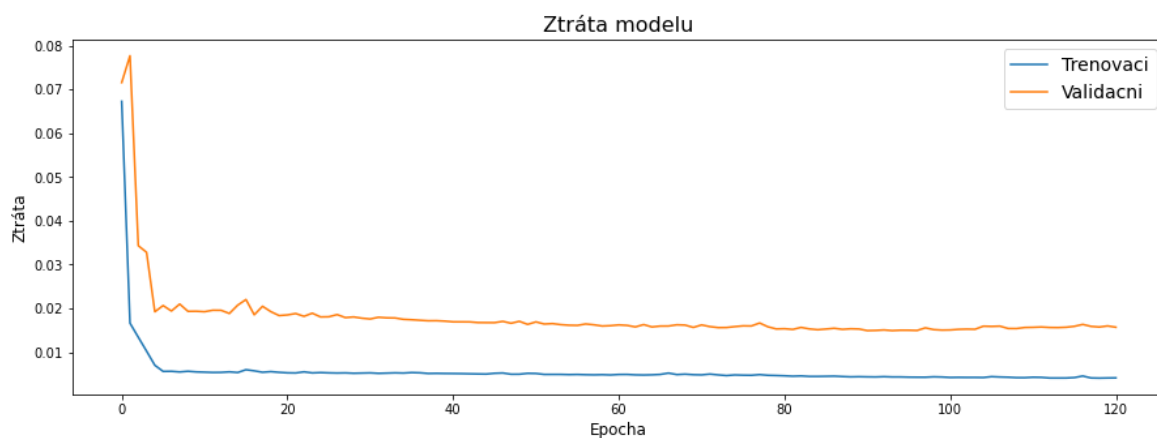
Graf 21: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 16-64-32-1



Zdroj: vlastní zpracování

4.4.1.4 Čtvrtá simulace pro konfiguraci sítě 32-64-128-1 a 32-128-64-1

Graf 22: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 32-64-128-1



Zdroj: vlastní zpracování

Pokud je dobrá predikční schopnost modelu posuzována, museli by trénovací a validační křivka překrývat. Podle Graf 22 jsou ale tyto funkce od se velmi vzdálené. Zatím nejhůř z předešlých. Dalo by se tedy uvažovat nad tím, že se model nemůže, naučit trénovacímu souboru dat. Nicméně graf je i tak spíše podobný grafům s uspokojivou predikční schopností. Funkce předčasného ukončení byla zaktivována i v tomto případě a to po 91 epoše. Výpočet končí po 121 epochách.

Graf 23: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 32-64-128-1

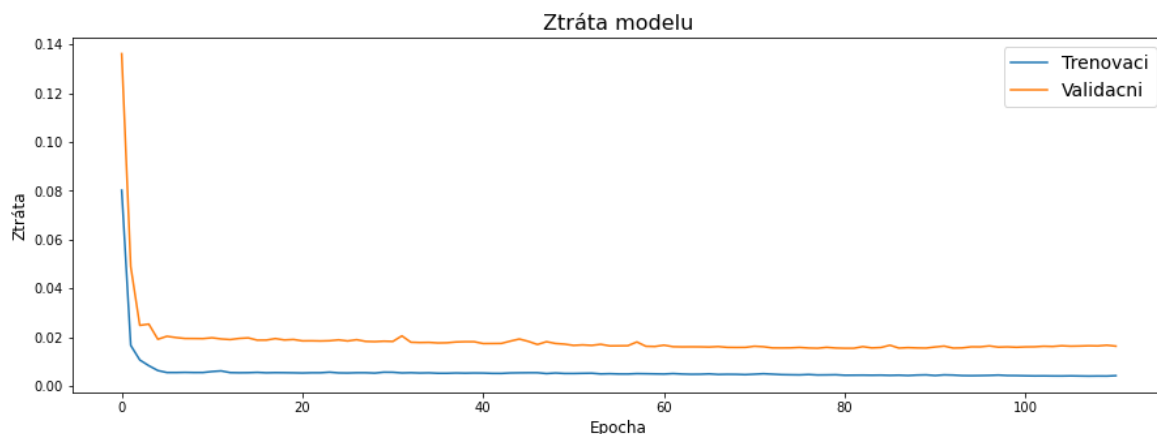


Zdroj: vlastní zpracování

Minimální hodnota časové řady pro predikovanou hodnotu AOV je 40,57 USD. Maximální je 67,68 USD. Směrodatná odchylka je 5,10 USD. RMSE je v tomto případě zatím nejhorší, a to 7,9893 USD.

Poslední konfigurací je 32-128-64-1. Její grafické vyjádření průběhu trénování a validace je znázorněna na grafu 24.

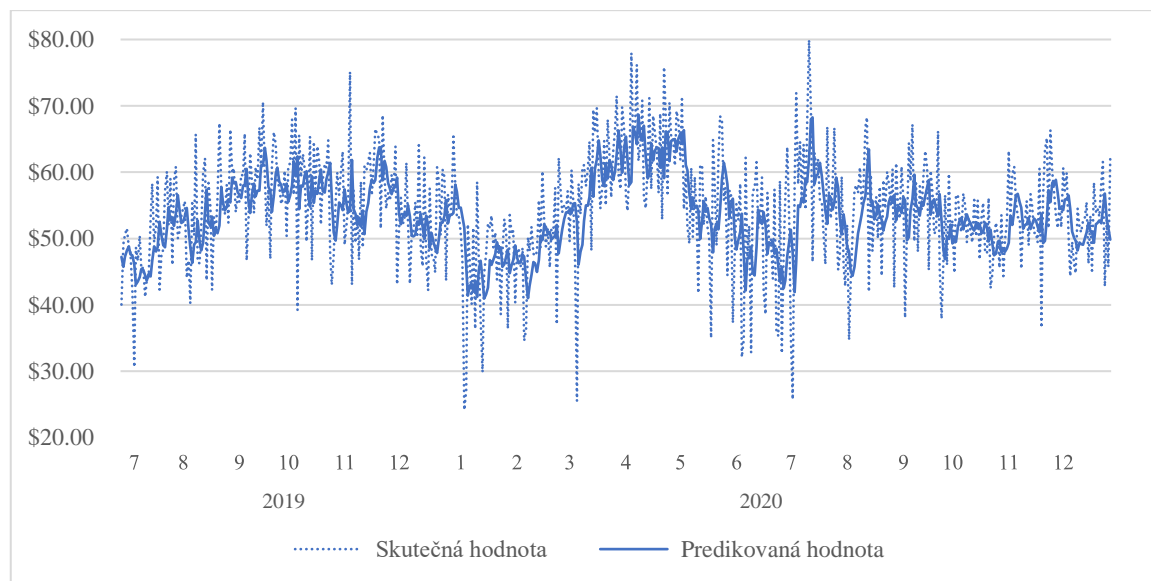
Graf 24: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě 32-128-64-1



Zdroj: vlastní zpracování

Stejně tak jako u konfigurace sítě 32-64-128-1 je rozdíl je mezi validační a trénovací křivkou v poměru s ostatními modely zvýšený. Model lze prezentovat jako model s uspokojivou predikční schopností. Interace skončila po 121 epochách.

Graf 25: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě 32-128-64-1



Zdroj: vlastní zpracování

RMSE modelu je 7,9347 USD. Maximum nalezené v souboru dat s predikovanými hodnotami je 40,95 USD, skutečné maximum je pak 68,67 USD. Průměr je přibližně 53,62 USD.

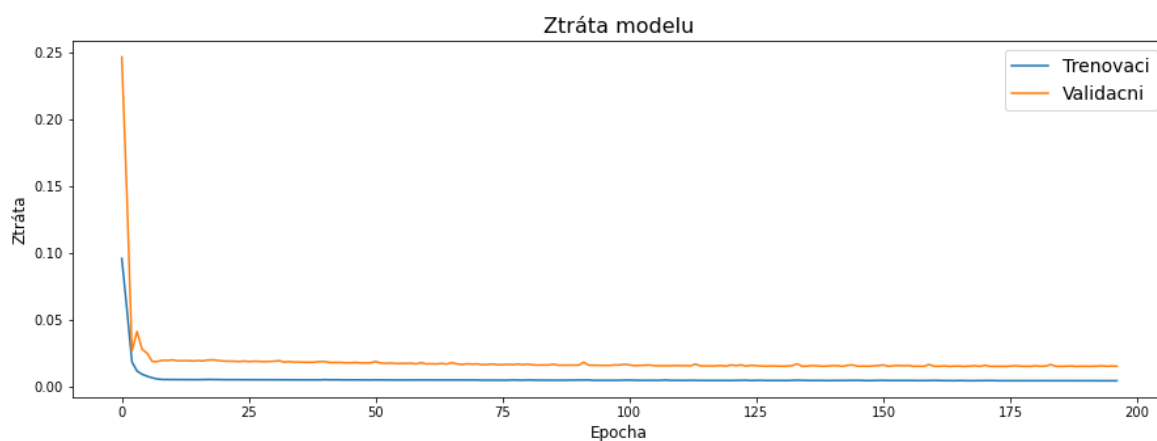
4.4.2 Pozměněná konfigurace sítě

Z předešlých simulací byla přijatá jako nejlepší varianta s konfigurací sítě 8-32-16-1, kde došlo k uspokojivé predikční schopnosti a jejíž RMSE bylo nejnižší, čili 7,8279 USD.

Tato nejlepší varianta je následně podrobena dalším dvěma simulacím, aby se zjistila možnost potenciálních lepších prediktivních výsledků, pokud by došlo k ubrání a přidání skryté vrstvy sítě. V prvním případě došlo k odebrání jedné ze skrytých vrstev. V případě druhém pak k jejímu přidání. Mohou nastat tři hlavní scénáře:

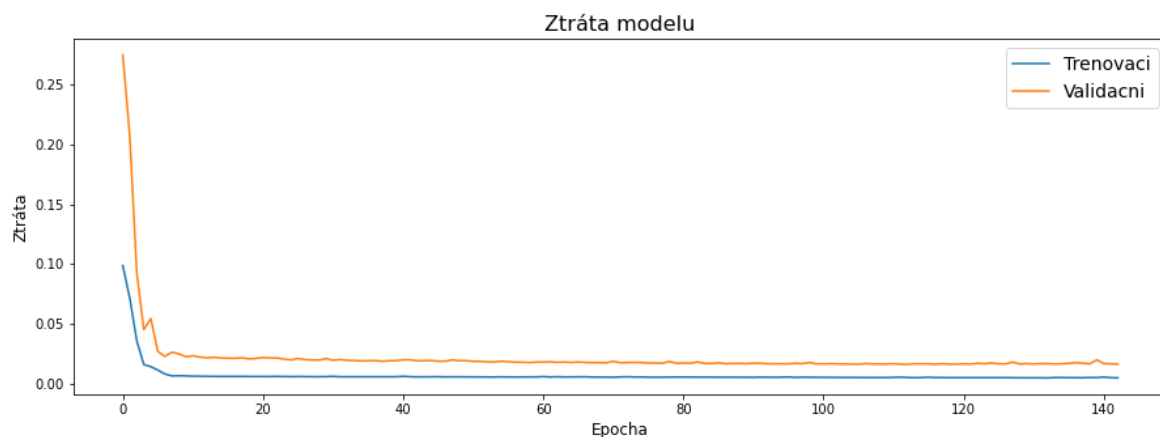
1. Buďto dojde k uspokojivé predikční schopnosti a zároveň hodnota RMSE bude vyšší než v případě původního modelu, pak model nebude přijat.
2. Model je nepoužitelný pro predikci, respektive má špatnou predikční schopnost. Potom model taktéž nebude přijat.
3. Model je přijat jako nejlepší, pokud splňuje podmínku uspokojivé predikční schopnosti a nižšího RMSE.

Graf 26: Znárodnění ztráty modelu v průběhu epoch s přidáním jednou skrytou vrstvou



Zdroj: vlastní zpracování

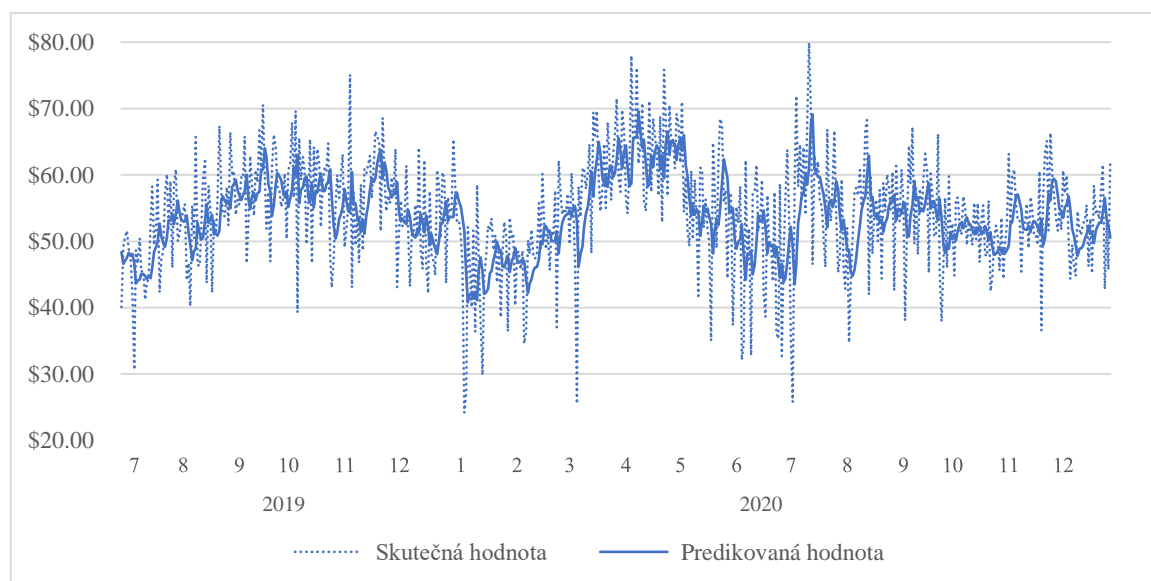
Graf 27: Znáznornění ztráty modelu v průběhu epoch pro konfiguraci sítě s ubranou jednou skrytou vrstvou



Zdroj: vlastní zpracování

Podle grafů 26 a 27 mají modely uspokojivé predikční schopnosti. Rozhodující je proto hodnota RMSE, kde v případě prvního modelu s konfigurací 8-32-1 je tato hodnota 7,8899 USD a pro model s konfigurací 8-32-16-8-1 je 7,9644 USD. V původním modelu s konfigurací 8-32-16-1 bylo RMSE rovno 7,8279 USD. To znamená, že ani jeden z modelů nedosahuje lepších predikčních schopností jako ten původní. Dále je patrné, že navýšení o jednu vrstvu zapříčinilo zhoršení modelu, a to až o 2 %. V obou případech trénování modelu došlo k předčasnému ukončení. První byl ukončen ve 197. epoše a druhý ve 143.

Graf 28: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě s přidanou jednou skrytou vrstvou



Zdroj: vlastní zpracování

Graf 29: Porovnání skutečné a predikované hodnoty pro konfiguraci sítě s ubranou jednou skrytou vrstvou



Zdroj: vlastní zpracování

U grafů Graf 28 a Graf 29 jsou patrné predikční schopnosti modelu. V prvním případě modelu je původní funkce daleko lépe aproximována. Interval výskytu hodnot AOV je pro síť bez jedné vrstvy od 40,79 USD do 69,56 USD. U druhé sítě je mezi 39,88 USD a 66,23 USD. Interval je pro síť s konfigurací 8-32-16-8-1 větší o více než 2 USD. Tato síť má i menší průměr AOV, a to 53,46 USD, u první je průměr 53,78 USD.

5 Výsledky a diskuze

5.1 Výsledky základní analýzy dat

Pro účely této práce byl vybrán jeden specifický druh kupónu, u kterého dochází k největším výkyvům v čase. Jedná se o především o kupóny nabízející roční licence v USA.

5.1.1 Transakce a tržby

Podle evidovaných záznamů je patrný dlouhodobý trend poklesu transakcí a tržeb. O tom vypovídá i snížení počtu transakcí za den. V roce 2016 byl průměr 708 transakcí, zatímco v roce 2020 už jenom 107 transakcí denně. Pokud se tedy celkové transakce v roce 2016 rovnaly 259 tisícům zakoupených licencí a v roce 2020 pouze 39 tisícům zakoupených licencí, jednalo se tak o pokles o skoro 85 %. Obdobně tomu je i u tržeb, kde došlo ke snížení v průběhu let až o 76 %, čili z 8,84 milionů USD za rok 2016 na 2,12 milionů USD za rok 2020. V roce 2019 došlo ke stagnaci jednak ze strany tržeb, tak i počtu transakcí. V roce 2020 je pak patrný trend růstu především u počtu tržeb.

5.1.2 AOV

V návaznosti na předchozí zjištění je pozorován i nárůst hodnot AOV v průběhu let. Nejvíce tomu je například v roce 2018, kde došlo k nejvýraznějšímu nárůstu. V roce 2016 byla průměrná hodnota AOV 34 USD, oproti tomu v roce 2020 byla už 54 USD. To je nárůst o 36 %. V letech 2019 a 2020 došlo kromě sezónních výkyvů ke stabilizaci AOV a rostoucí trend dále nepokračoval.

5.1.3 Faktory ovlivňující cenu kupónu

Na cenu kupónu mohou mít vliv dva hlavní faktory. Tím prvním je strategie prodeje specifických licencí a změna jejich cen, dále pak konkurenční prostředí.

Co se týče strategie prodeje kupónů, je nutné si uvědomit, že po celou dobu bylo pracováno s časovou řadou obsahující pouze data za jeden kupón. V průběhu let tak mohlo dojít ke změně strategie propagace a změny slev na tento druh kupónu, které vedly k nárůstu prodeje jiných druhů licencí (např. tříletých apod.). I výrazná část poskytovaných promo akcí je v posledních letech orientována na prodej těchto kupónů. O dané situaci vypovídá i růst tržeb a transakcí v průběhu roku 2020. Zde docházelo k značnému posílení promo akcí i na ročních licencích. Nárůst je možný i z důvodů prodeje specifických modifikací promo

akcí na ročních kupónech, jako jsou například kampaně nabízející určitý počet měsíců navíc jako „dárek“ (například kampaň 12 + 3 měsíce zdarma).

Dalším faktorem ovlivňujícím cenu kupónu je konkurenční prostředí. Ačkoli je tato VPN velmi technologicky vyspělá a každým rokem je oceňována jako jedna z nejlepších desítky VPN na světě (Top 10 VPN, 2021), je nucena se trhu s VPN přizpůsobovat. Strach ze ztráty zákazníka vede firmy k promování dlouhodobých licencí. Tyto aspekty ale vedly ke zvyšování cen u ročních kupónů, a naopak nabízení velkých výprodejů na straně kupónů dvouletých či tříletých. Zákazníci mají pak tendenci u obnovování licence konvertovat k časově delším a levnějším nabídkám.

V neposlední řadě by neměla zůstat opomenuta specifická situace, která od prvního kvartálu roku 2020 měnila chování a potřeby všech spotřebitelů. Samozřejmě jde o celosvětovou pandemii onemocnění SARS-CoV-2, jež díky vládním nařízením u většiny států došlo k prudkému nárůstu práce z domova. Tato situace vedla firmy i jedince k potřebě zabezpečení svých komunikačních zařízení. O tom vypovídá i růst tržní hodnoty společnosti Avast, která je jednou z největších firem na světě a která nabízí celou řadu služeb pro ochranu v online prostředí (Krejčí, 2020).

5.2 Výsledky modelu LSTM

Po několika zkušebních výpočtech v prostředí Microsoft Machine Learning Studio bylo rozhodnuto ustoupit od tohoto nástroje. Důvodem pak byly především vysoké hodnoty RMSE, kde nejnižší možná dosažená hodnota byla 9,8599. Díky tomu byla změněna strategie v používání jiných nástrojů pro zlepšení predikční schopnosti modelů.

Pro další práci byl vybrán model Long short term memory (LSTM), který má základ v rekurentních neuronových sítích, jenž díky své struktuře umožňuje zapamatování i předešlých hodnot. To je užitečné především v časových řadách, kde je hodnota závislá na vývoji v předešlých dnech.

5.2.1 Výsledky simulace modelu LSTM

Jaký z vypočtených modelů neuronových sítí je nejvhodnější, je možné poznat jednak díky RMSE ale také z posouzení grafu vzhledem k vyhodnocení predikční schopnosti. V případě RMSE bude uvažována minimální hodnota a vzhledem k predikční schopnosti je pak považována uspokojivá predikční schopnost modelu.

Tabulka 22: Přehled výsledků simulací

Model	Simulace	Konfigurace sítě	Predikční schopnost	RMSE
1	První	4-8-16-1	Uspokojivá	7,9469
2	První	4-16-8-1	Uspokojivá	7,8376
3	Druhá	8-16-32-1	Uspokojivá	7,9801
4	Druhá	8-32-16-1	Uspokojivá	7,8279
5	Třetí	16-32-64-1	Uspokojivá	7,8909
6	Třetí	16-64-32-1	Uspokojivá	7,9102
7	Čtvrtá	32-64-128-1	Uspokojivá	7,9038
8	Čtvrtá	32-128-64-1	Uspokojivá	7,9347
9	Pátá	8-32-1	Uspokojivá	7,8279
10	Pátá	8-32-16-8-1	Uspokojivá	7,9644

Zdroj: vlastní zpracování

Uspokojivá schopnost modelu byla evidována ve všech simulacích. K největším rozdílům mezi validačními a trénovacími křivkami docházelo u sítí, jejichž počet neuronů ve skrytých vrstvách byl nejvyšší. To se týká konfigurací sítě 32-64-128-1 a 32-128-64-1, respektive sítí ze čtvrté simulace.

Finální rozhodnutí o nejlepším možném modelu je tedy dáno pomocí hodnot RMSE. Tato hodnota se běžně používá právě k vybrání toho nejlepšího modelu. Čím je menší RMSE, tím lépe predikovaná data aproximují původní soubor dat. Minimální hodnota je pro konfiguraci sítě 8-32-16-1, kde se hodnota AOV blíží 7,83 USD. Je zajímavé, že ale obdobná konfigurace sítě jen s prohozenými skrytými vrstvami, disponuje zase maximální hodnotou a to 7,98 USD. Čili druhá simulace je nejlepší možnou ale zároveň i nejhorší. To vyplývá o tom, jak je konfigurace sítě a počet neuronů v jednotlivých sítích důležitý. Obecně se hodnota RMSE v průměru pohybuje okolo 7,90 USD a žádný z modelů nepřesáhl hodnotu 8. Směrodatná odchylka je pouhých 0,06 USD. Proto se dá předpokládat, že všechny modely predikují hodnoty podobné. Avšak pro další postup byl logicky vybrán model nejlepší.

5.2.2 Sestavení modelu pro tvorbu ceny v prostředí online prodeje

Díky sekvenčnímu uspořádání dat a použitému algoritmu výpočtů pomocí LSTM modelu, lze tedy předpovědět jednu hodnotu v čase $t+1$ na základě sedmi předcházejících dní. Toto predikované AOV se stane centrem pro další uvažování o stanovení ceny. Jak bylo zmíněno v metodické části, s cenami produktu VPN je pracováno ve více odděleních jako je podpora a e-commerce.

5.2.2.1 Oddělení podpory

Oddělení podpory komunikuje přímo se zákazníky. Samo si stanovuje ceny kupónů. Zmiňovaná subjektivita značně ovlivňuje celkové tržby a také denní výkyvy v AOV (podrobněji popsáno v kapitole 2.2). Nabízená cena, stanovená podle predikovaného AOV, jako náhrada za vzniklé případné komplikace, by nemusela vést k dostatečným kompenzacím vzhledem k frustraci zákazníka. Proto je nutné tuto subjektivitu do výpočtu ceny pro oddělení podpory zakomponovat.

V průběhu výpočtů se objevil i jeden velmi zajímavý úkaz, který by do jisté míry mohl pomoci ve stabilizaci rozptylu AOV a tím i snížení maximálních a minimálních hodnot, která jsou patrná ve všech případech simulací. Zejména pak pro model s nejnižším RMSE je minimální hodnota přibližně 37,67 USD a maximální přibližně 67,42 USD. V případě původního souboru je minimální hodnota o 13 USD nižší a maximální o 12 USD vyšší. Toho by se dalo využít i u stanovení intervalu AOV, na základě kterého by se tým podpory mohl rozhodovat. Vždy podle situace by pak odvíjel cenu od predikovaného AOV ve stanoveném intervalu.

V neposlední řadě by sleva vytvořená z predikovaného AOV vytvořená týmem podpory měla být porovnána s minimální hraniční cenou, za kterou se smí prodávat tento produkt (dáno například náklady na produkt a jiné). Tato cena by měla být pevně stanovena a zakomponována do strategie prodeje kupónů.

5.2.2.2 Oddělení e-commerce

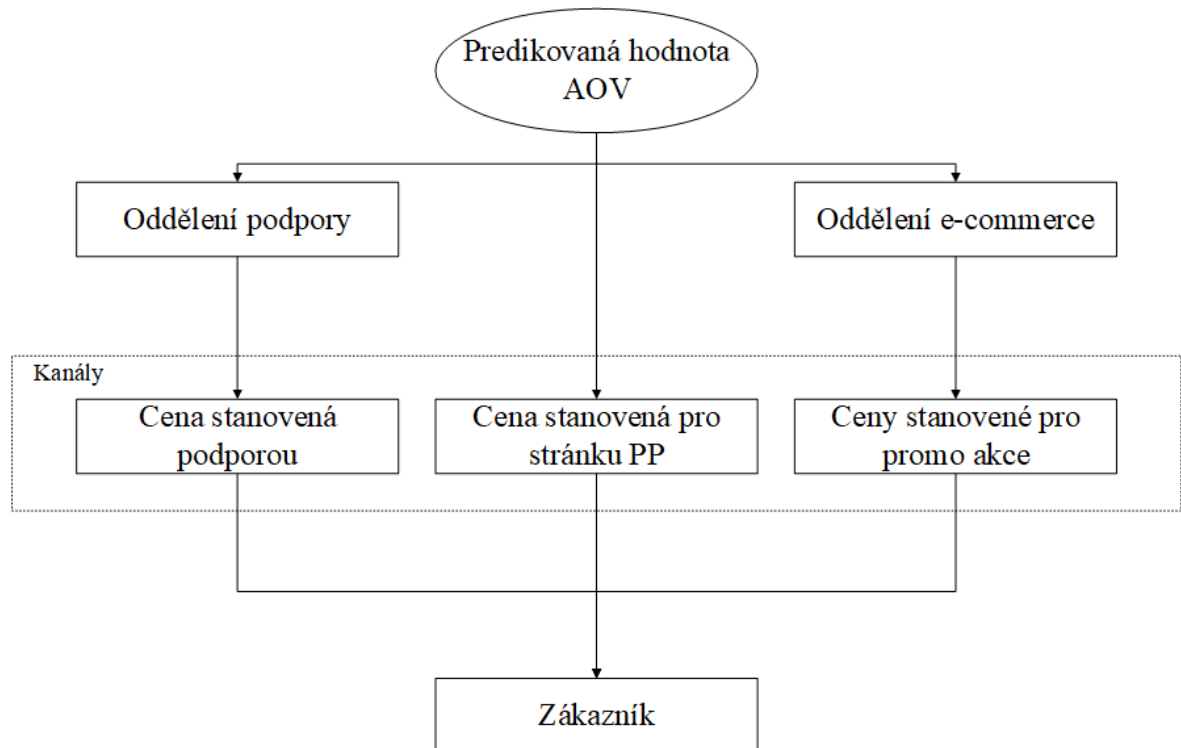
Oddělení e-commerce pracuje především s kanály webového messagingu (dále WM), hero a pricing page (dále PP), což bylo podrobněji popsáno v kapitole 2.1.. Prezentování cen pak může být provedeno dvěma způsoby.

1. Na PP se většinou prezentují kupóny bez slev. Predikovaná hodnota AOV by zde tedy mohla být bez jakýchkoliv dalších úprav.

2. V případě promování akcemi, které se děje především v kanálech webového messagingu a hero. Tady by mohlo docházet ke stanovení slevy, která by následně byla vypočítána z predikované hodnoty AOV.

Tato by, dle výše uvedeného, mohlo vypadat schéma pro tvorbu ceny prezentovanou finálnímu zákazníkovi ve vztahu k jednotlivým kanálům.

Obrázek 18: Schéma nastavení cen pro finálního zákazníka



Zdroj: vlastní zpracování

6 Závěr

V práci se autor zaměřil na řešení problému tvorby ceny v prostředí e-commerce. Cílem práce bylo vytvoření postupu dynamické cenotvorby pro online prodej nehmotného produktu za účelem maximalizace tržeb. Tento cíl se dle názoru autora podařilo naplnit. Na základě teoretické rešerše byla navržena vhodná metodika pro vlastní práci. V té byly soubory dat nejprve podrobeny rozboru, při kterém došlo například k zjištění, že se celkové prodeje jednoročních kupónů v průběhu let snižovaly, nebo že docházelo ke zdražování ceny ročních kupónů. Také bylo zjištěno, že se v minulém roce trend dlouhodobého poklesu tržeb otočil a nyní roste. To autor přisoudil mnoha různým činitelům. Jednou je změna strategie prodeje kupónů ale také zvýšená poptávka po produktu. Následně data byla zformulována do požadované podoby pro práci s modelem Long-short term memory (LSTM), který umí pracovat se sekvenčními daty.

Na simulacích bylo předvedeno, jak předem zvolená konfigurace může ovlivňovat trénování sítě. Vzhledem k tomu, že ve všech případech modely vykazovaly uspokojivou predikční schopnost, nejlepší byl zvolen na základě hodnoty RMSE, konkrétně model s konfigurací sítě 8-32-16-1.

Na základě tohoto modelu byly následně predikované hodnoty AOV interpretované pro praktické použití pro rozhodování zaměstnanců v odděleních e-commerce a podpory. Právě pro ně byly navrženy jednoduché postupy, jak cenu stanovit. U týmu podpory byl brán v úvahu například individuální přístup k zákazníkovi podle míry jeho frustrace. Kolem predikované hodnoty AOV byl navržen interval, který umožňuje potřebnou flexibilitu pro řešení těchto situací. V kanálech promo akcí je doporučeno predikovanou hodnotu AOV násobit procentuální stanovenou slevou pro danou akci.

Jedním z vypočítaných faktů byl ten, že se dopočítané tržby u souboru s aktuálními a prediktivními hodnotami liší jen málo. Zároveň byla v prediktivních datech značně snižena fluktuace oproti původnímu souboru.

Pro budoucí zkoumání se autor doporučuje zaměřit na porovnání všech typů kupónů a ty následně porovnávat. Za zmínku stojí i zvážení konkurenčních produktů, u kterých by měl být rovněž dlouhodobě sledován vývoj jejich cen.

7 Seznam použitých zdrojů

AIWorld, 2019. *Co je Keras? Vysvětlujeme aplikační rozhraní pro hluboké neuronové sítě.* [Online]

Available at: <https://aiworld.cz/umela-inteligence/co-je-keras-vysvetlujeme-aplikacni-rozhрани-pro-hluboke-neuronove-site-224>

[Přístup získán 10 Březen 2021].

Arbel, N., 2018. *How LSTM networks solve the problem of vanishing gradients.* [Online]

Available at: <https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577>

[Přístup získán 2 Únor 2021].

Barnston, A. G., 1992. *Correspondance among the Correlation, RMSE, and Heidke Forecast Verification Measures.* [Online]

Available at:

<https://www.swpc.noaa.gov/sites/default/files/images/u30/Barnston,%20Anthony%20G.,%201992.pdf>

[Přístup získán 10 Březen 2021].

Biskup, R., 2009. *Možnosti neuronových sítí.* Disertační práce editor Praha: Česká zemědělská univerzita v Praze.

Brownlee, J., 2019. *How to use Learning Curves to Diagnose Machine Learning Model Performance.* [Online]

Available at: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>

[Přístup získán 20 Březen 2021].

Brownlee, J., 2019. *How to use Learning Curves to Diagnose Machine Learning Model Performance.* [Online]

Available at: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>

[Přístup získán 15 Leden 2021].

Cho, K. a další, 2014. *Learning Phrase Representations using RNN Encoder–Decoder.* [Online]

Available at: <https://arxiv.org/pdf/1406.1078v3.pdf>

[Přístup získán 10 Leden 2021].

Contants, 2018. *Average Order Value and its importance to online Ecommerce business.*

[Online]

Available at: <https://medium.com/@Countants/average-order-value-and-its-importance-to-online-ecommerce-business-e0fd3f06c56e>

[Přístup získán 10 February 2021].

Evropský parlament, 2020. *Co je umělá inteligence a jak ji využíváme?.* [Online]

Available at:

<https://www.europarl.europa.eu/news/cs/headlines/society/20200827STO85804/umela-intelligence-definice-a-vyuziti#:~:text=Definice%20um%C4%9B1%C3%A9%20inteligence,u%C4%8Den%C3%AD%20se%2C%20pl%C3%A1nov%C3%A1n%C3%AD%20nebo%20kreativita.>

[Přístup získán 28 Prosinec 2020].

Helios, 2017. *Vše, co jste chtěli vědět o strojovém učení.* [Online]

Available at: <https://blog.helios.eu/cz/clanky/vse-co-jste-chteli-vedet-o-strojovem-uceni/>

[Přístup získán 12 Listopad 2021].

Jain, P., 2019. *Complete Guide of Activation Functions.* [Online]

Available at: <https://laptrinhx.com/complete-guide-of-activation-functions-574622854/>

[Přístup získán 20 Březen 2021].

Jourová, V., 2016. *Dynamická tvorba cen a spotřebitelé v EU.* [Online]

Available at: https://www.europarl.europa.eu/doceo/document/E-8-2016-002800-ASW_CS.html

[Přístup získán 10 Leden 2021].

Jupyter, 2021. *The Jupyter Notebook.* [Online]

Available at: <https://jupyter.org/>

[Přístup získán 10 Březen 2021].

Kasík, P., 2014. *Počítač přesvědčivě napodobil člověka. Hrál si na pubertáka.* [Online]

Available at: [https://www.idnes.cz/technet/software/uspesny-turingu-
test.A140609_181110_software_pka](https://www.idnes.cz/technet/software/uspesny-turingu-
test.A140609_181110_software_pka)

[Přístup získán 28 Prosinec 2020].

Koval, V., 2020. *Český rozhlas Radiožunál.* [Online]

Available at: <https://radiozurnal.rozhlas.cz/jsme-lepsi-nez-si-myslme-praha-mohla-byt-jako-silicon-valley-tvrdi-propagatorka-8246661>

[Přístup získán 10 Leden 2021].

- Krejčí, J., 2020. *Avast těží z přechodu světa na home office. Jeho akcie rostou, tržní hodnota firmy je nyní téměř 180 miliard korun.* [Online]
Available at: <https://archiv.ihned.cz/c1-66802030-avast-tezi-z-prechodu-sveta-na-home-office-jeho-akcie-rostou-trzni-hodnota-firmy-je-nyni-temer-180-miliard-korun>
[Přístup získán 20 Březen 2021].
- Křivý, I., 2012. *Analýza časových řad*, Ostrava: Ostravská univerzita v Ostravě.
- Matplotlib, 2021. *Matplotlib: Visualization with Python.* [Online]
Available at: <https://matplotlib.org/>
[Přístup získán 10 Březen 2021].
- Ministerstvo průmyslu a obchodu, 2019. *Národní strategie umělé inteligence v České republice*, místo neznámé: autor neznámý
- NumPy, 2021. *ABOUT US.* [Online]
Available at: <https://numpy.org/about/>
[Přístup získán 10 Březen 2021].
- Olah, C., 2015. *Understanding LSTM Networks.* [Online]
Available at: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
[Přístup získán 21 Prosinec 2020].
- Olah, C., 2015. *Understanding LSTM Networks.* [Online]
Available at: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
[Přístup získán 2 Únor 2021].
- Pandas, 2021. *About pandas.* [Online]
Available at: <https://pandas.pydata.org/about/>
[Přístup získán 10 Březen 2021].
- Pohl, M., 2019. *Iniciativa prg.ai udělá z Prahy centrum umělé inteligence.* [Online]
Available at: <https://www.avcr.cz/cs/o-nas/aktuality/Iniciativa-prg.ai-udela-z-Prahy-centrum-umele-inteligence/>
[Přístup získán 22 Leden 2021].
- Sadawi, N., 2020. *Advanced Machine Learning: Create Real-World ML Projects.* [Online]
Available at: <https://learning.oreilly.com/live-training/courses/advanced-machine-learning-create-real-world-ml-projects/0636920478799/>
[Přístup získán 22 Prosinec 2020].
- Samir, R. & Udit, C., 2013. *Soft Computing.* 1 editor Indie: Pearson Education.

- Siddique, N. & Adeli, H., 2013. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*. USA: John Wiley.
- TensorFlow, 2021. *Why TensorFlow*. [Online]
Available at: <https://www.tensorflow.org/about>
[Přístup získán 10 Leden 2021].
- Top 10 VPN, 2021. *Top 10 Best VPNs for March*. [Online]
Available at:
https://www.top10vpn.com/top10/?v=header&bsid=bndse1kw001&gclid=Cj0KCQjwmIuDBhDXARIsAFITC_7VOhPPMuwtNhIAVaPPdBhmLmbFSLc_Hj7AydWjhij8b0NofvZGwFAaAlqbEALw_wcB
[Přístup získán 20 Březen 2021].
- TriloByte, 2021. *Neuronová síť*. [Online]
Available at: https://www.trilobyte.cz/downloadfree/qcemanual/neural_net.pdf
[Přístup získán 20 Leden 2021].
- Tučková, J., 2003. *Úvod do teorie a aplikací umělých neuronových sítí*. Praha: ČVUT.
- Tučková, J., 2009. *Vybrané aplikace umělých neuronových sítí při zpracování signálů*. 1 editor Praha: České vysoké učení technické v Praze.
- Turing, A. M., 1950. Computing Machinery and Intelligence. *JSTOR*, 59(236), pp. 433-460.
- Tvrď, F., 2011. *Turingův test*. [Online]
Available at: https://theses.cz/id/st6pmz/Turingv_test.pdf
[Přístup získán 28 Prosinec 2020].
- Veselý, A., 2005. *Artificial intelligence*. 1 editor Praha: Česká zemědělská univerzita v Praze Provozně ekonomická fakulta.
- Veselý, A., 2005. *Úvod do umělé inteligence*. 1 editor Praha: Česká zemědělská univerzita v Praze Provozně ekonomická fakulta.
- Veselý, A., 2012. *Metody umělé inteligence*. 1 editor Praha: Česká zemědělská univerzita v Praze Provozně ekonomická fakulta.
- Volná, E., Kotyrba, M., Janošek, M. & Kocian, V., 2014. *Umělá inteligence - Rozpoznávání vzorů v dynamických datech*. 1 editor Praha: BEN - technická literatura.
- Yagcioglu, S., 2020. *Classical Examples of Supervised vs. Unsupervised Learning in Machine Learning*. [Online]
Available at: <https://www.springboard.com/blog/lp-machine-learning-unsupervised->

learning-supervised-learning/

[Přístup získán 5 February 2021].

Zichová, J., 2021. *LINEÁRNÍ REGRESNÍ MODEL*. [Online]

Available at:

<https://www2.karlin.mff.cuni.cz/~zichova/PRFUK/MatStat/LINE%C3%81RN%C3%8D%20REGRESN%C3%8D%20MODEL.pdf>

[Přístup získán 10 Leden 2021].

8 Přílohy